MySQL 5.7 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 5.7, up through MySQL 5.7.5. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 5.7 documentation, see the MySQL 5.7 Reference Manual, which includes an overview of features added in MySQL 5.7 (What Is New in MySQL 5.7), and discussion of upgrade issues that you may encounter for upgrades from MySQL 5.6 to MySQL 5.7 (Upgrading from MySQL 5.6 to 5.7).

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (http://dev.mysql.com/downloads/), it means that the version has not yet been released.

The date mentioned with a release version is the date of the last revision control system changeset on which the release was based, not necessarily the date when the distribution packages were made available. The binaries are usually made available a few days after the date of the tagged changeset because building and testing all packages takes some time.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

For legal information, see the Legal Notices.

Document generated on: 2014-03-22 (revision: 3045)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 5.7, up through MySQL 5.7.5.

Legal Notices

For more information on the terms of this license, or for details on how the MySQL documentation is built and produced, please visit MySQL Contact & Questions.

For additional licensing information, including licenses for third-party libraries used by MySQL products, see Preface and Legal Notices.

For help with using MySQL, please visit either the MySQL Forums or MySQL Mailing Lists where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the MySQL Documentation Library.

Changes in MySQL 5.7.5 (Not yet released, Milestone 15)

Note

This is a milestone release, for use at your own risk. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward.

Functionality Added or Changed

- Previously, the `MYSQL_MAINTAINER_MODE` `CMake` option was turned on by default for debug builds and off for release builds, and `MYSQL_MAINTAINER_MODE` caused `-Werror` to be enabled when building with GCC. This made it cumbersome to enable `-Werror` under certain conditions, such as when compiling with Clang.

  Now, `MYSQL_MAINTAINER_MODE` is on by default when compiling debug builds with GCC, and `MYSQL_MAINTAINER_MODE` enbles `-Werror` regardless of whether GCC or Clang is used. Enabling `-Werror` with Clang can be done simply by explicitly setting `-DMYSQL_MAINTAINER_MODE=1` when running `CMake`. In addition, some compilation warnings reported by Clang 3.4 were fixed, making it possible to build the default debug build with `-Werror`. (Bug #18313717)

- Optimizer trace output for range access in the `considered_access_path` section has been improved: Instead of always printing `"access_type": "ref"` for index lookup types, `"eq_ref"`, `"ref"`, or `"fulltext"` is now printed. (Bug #18195373)

- The obsolete and unmaintained `charset2html` utility has been removed from MySQL distributions. (Bug #71897, Bug #18352347)

- A new status variable, `Max_used_connections_time`, indicates the time at which `Max_used_connections` reached its current value. Thanks to Jordi Prats for the patch. (Bug #59738, Bug #11766596)

Bugs Fixed

- **Performance; Replication:** When processing the dump thread, a semisynchronous replication master checked whether or not the dump thread came from a semisynchronous slave by checking the value of rpl_semi_sync_slave_enabled, but did so for every operation performed on this thread, which had significant negative impact on performance. Now this check is made only once, when the dump thread is started, which should noticeably improve the performance of semisynchronous replication in most cases. (Bug #17932935)
- **InnoDB:** The os_event_wait_time_low function would sometimes return OS_SYNC_TIME_EXCEEDED before the sync time has elapsed. (Bug #18386498)
- **InnoDB:** Running a SELECT on a partitioned table caused a memory access violation in memcpy(). (Bug #18383840)
- **InnoDB:** Added the C++ ostream mechanism for error logging. (Bug #18320915)
- **InnoDB:** Code quality improvements for the redo log subsystem. (Bug #18345004)
- **InnoDB:** Removed the recv_max_parsed_page_no code variable, which was only used in a diagnostic error message. (Bug #18312967)
- **InnoDB:** Calls to sched_getcpu would cause page faults. (Bug #18225489)
- **InnoDB:** ib_heap_resize failed to verify that new_size is greater than or equal to old_size before calling memcpy. (Bug #18178915)
- **InnoDB:** Setting innodb_max_dirty_pages_pct=0 would leave 1% of dirty pages unflushed. Buffer pool flushing is initiated when the percentage of dirty pages is greater innodb_max_dirty_pages_pct. The internal variables that store the innodb_max_dirty_pages_pct value and the percentage of dirty pages (buf_get_modified_ratio_pct and srv_max_buf_pool_modified_pct) were defined as unsigned integer data types, which meant that a innodb_max_dirty_pages_pct value of 0 required a dirty pages percentage of 1 or greater to initiate buffer pool flushing.

  To address this problem, the buf_get_modified_ratio_pct and srv_max_buf_pool_modified_pct internal variables are redefined as double data types, which changes the range value for innodb_max_dirty_pages_pct and innodb_max_dirty_pages_pct_lwm from 0 .. 99 to 0 .. 99.99. Additionally, buffer pool flushing is now initiated when the percentage of dirty pages is "greater than or equal to" innodb_max_dirty_pages_pct. (Bug #13029450)
- **Partitioning:** When the index_merge_intersection flag (enabled by default) or the index_merge_union flag was enabled by the setting of the optimizer_switch system variable, queries returned incorrect results when executed against partitoned tables that used the MyISAM storage engine, as well as partitioned InnoDB tables that lacked a primary key. (Bug #18167648)

  References: See also Bug #16862316, Bug #17588348, Bug #17648468.
- **Replication:** During relay log initialization, the thread context was used as a flag for the reconstruction of the retrieved GTID set, an operation that does not depend on this parameter. This could be problematic if relay log initialization was called in another context other than the legacy replication scenario; if the invocation was made in a context where the thread context was always present, this prevented the set's reconstruction. The opposite could also happen when the thread context was not present, which cause the initialization to be performed twice.

  To avoid such issues, the thread context flag is replaced with a new flag that allows the reconstruction in all contexts but prevents multiple invocations. (Bug #18337036)

- **Replication:** In certain cases, the server mishandled triggers and stored procedures that tried to modify other tables when called by `CREATE TABLE ... SELECT`. This is now handled correctly as an error. (Bug #18137535)
- **Replication:** When `gtid_mode=ON`, and a transaction is filtered out on the slave, the GTID of the transaction is still logged on the slave as an "empty" transaction (consisting of a GTID followed immediately by `BEGIN` and then `COMMIT`). This is necessary to prevent the transaction from being retransmitted the next time the slave reconnects or is involved in a failover. The current fix addresses two issues relating to such "empty" transactions:

  - No empty transaction was generated for `CREATE TEMPORARY TABLE` or `DROP TEMPORARY TABLE` statements.
  - If the slave used a database filter (`--replicate-do-db` or `--replicate-ignore-db` option), no empty transaction was generated.

  (Bug #71326, Bug #18095502, Bug #18145032)
- Queries that use `AES_ENCRYPT()`, `AES_DECRYPT()`, or `RANDOM_BYTES()` cannot be cached but were permitted to enter the query cache. (Bug #18383169)
- `EXPLAIN FOR CONNECTION` showed an incorrect filtering value for dynamic range queries. (Bug #18395059)
- The `HAVE_IB_LINUX_FUTEX` `CMake` check failed due to a missing include file. (Bug #18353271)
- After a code reorganization in MySQL 5.7.4, `ORDER BY` for multiple-table `UPDATE` statements was ignored. (Bug #18352634)
- In debug builds, lack of proper object initialization of decimal objects caused an assertion to be raised. (Bug #18335446)
- Certain `INFORMATION_SCHEMA` queries could cause a server exit. (Bug #18319790)
- Solaris-specific scripts were included in and installed by non-Solaris packages. (Bug #18305641)
- For queries involving an `AND` of two geometry ranges, the optimizer could decide no index was usable but try to use it anyway, resulting in a server exit. (Bug #18220153)
- The C client library could leak memory when client plugins were used. (Bug #17933308)
- A `BEFORE UPDATE` trigger could insert `NULL` into a `NOT NULL` column. (Bug #17864349)
- A (rare) deadlock could occur between `LOCK_thd_data` and the `InnoDB` `trx_sys` mutex. One thread could read a query string while it was being removed by another thread. (Bug #17606098)
- `mysqldump` could create table definitions in the dump file that resulted in `Too many columns` errors when reloading the dump file. (Bug #17477959)
- Using `LOAD DATA INFILE` to load fixed-length data into a view could cause a server exit. (Bug #17458965)
- Row constructor arguments to `INTERVAL()` could cause a server exit. (Bug #16439419)
- When MySQL runs as service on Windows, `NTService.Stop()` initiates shutdown and exit events during shutdown. After a code reorganization in MySQL 5.7.3, a call to `clean_up()` was missed, resulting in initiation of crash recovery. (Bug #71104, Bug #17980260)
- Some statements could be written to the slow query log twice. (Bug #68626, Bug #16467055)

- LOAD DATA LOCAL INFILE could use all CPU if import errors occurred when there were no line delimiters. (Bug #51840, Bug #11759519)
- Deadlock could occur if three threads simultaneously performed INSTALL PLUGIN, SHOW VARIABLES, and mysql_change_user(). (Bug #71236, Bug #18008907)

Changes in MySQL 5.7.4 (Not yet released, Milestone 14)

> **Note**
>
> This is a milestone release, for use at your own risk. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running mysql_upgrade. For example, you may find it necessary to dump your data with mysqldump before the upgrade and reload it afterward.

Performance Schema Notes

- The Performance Schema now instruments prepared statements (for both the binary and text protocols):
  - Information about prepared statements is available in the prepared_statements_instances table. This table enables inspection of prepared statements used in the server and provides aggregated statistics about them.
  - The performance_schema_max_prepared_statements_instances system variable controls the size of the table.
  - The Performance_schema_prepared_statements_lost status variable indicates how many prepared statements could not be insrumented.

  For more information, see The prepared_statements_instances Table.

  If you upgrade to this release of MySQL from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate these changes into the performance_schema database.

Security Notes

- **Incompatible Change:** MySQL deployments installed using RPM packages now are secure by default. The following changes have been implemented as the default deployment characteristics:
  - The installation process creates only a single root account, 'root'@'localhost', automatically generates a random password for this account, and marks the password expired. The MySQL administrator must connect as root using the random password and use SET PASSWORD to select a new password. (The random password is found in the $HOME/.mysql_secret file.)
  - Installation creates no anonymous-user accounts.
  - Installation creates no test database.

  Those changes are implemented by modifying the default mode of operation for mysql_install_db, which is invoked automatically during RPM installation operations. Therefore, the changes also affect non-RPM installation methods for which you invoke mysql_install_db manually.

  Because mysql_install_db deployment now is secure by default, the --random-passwords option is unnecessary and has been removed. It has been replaced by the --skip-random-passwords option. You can use this option to explicitly produce a deployment that is *not* secure by default:

  - Installation creates a single root account, 'root'@'localhost', that has no password.
  - A test database is created that is accessible by any user.

  Note

  mysql_install_db no longer creates anonymous-user accounts, even with --skip-random-passwords.

- **Incompatible Change:** MySQL now enables database administrators to establish a policy for automatic password expiration: Any user who connects to the server using an account for which the password is past its permitted lifetime must change the password. The implementation has these components:

  - The default_password_lifetime system variable defines global password expiration policy. A value of *N* greater than zero means that passwords have a lifetime of *N* days. A value of 0 disables automatic password expiration. The default is 360; passwords must be changed approximately once per year.
  - The mysql.user table has new columns that store expiration policy information for individual accounts:

    - password_last_changed indicates when the password was last changed. The server uses this column when clients connect to determine whether the password is past its lifetime and must be changed per the expiration policy in effect.
    - password_lifetime indicates the account password lifetime. A value of *N* greater than zero means that the password has a lifetime of *N* days. 0 disables automatic password expiration. NULL (the default) means that the global expiration policy applies.
  - The ALTER USER statement has new options to set password expiration policy for individual accounts.

  If you upgrade to this release of MySQL from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate the changes to the mysql.user table. For each account, mysql_upgrade uses the time at which it executes to set the password_last_changed column and sets password_lifetime to NULL. This causes the default global expiration policy to go into effect (passwords have a lifetime of 360 days).

  For more information, see Password Expiration Policy, and ALTER USER Syntax.

Functionality Added or Changed

- **Performance; InnoDB:** InnoDB now supports multiple page_cleaner threads for flushing dirty pages from buffer pool instances. A new system variable, innodb_page_cleaners, is used to specify the number of page_cleaner threads. The default value of 1 maintains the pre-MySQL 5.7.4 configuration in which there is only a single page_cleaner thread. This enhancement builds on work completed in MySQL 5.6.2, which introduced a single page cleaner thread to offload buffer pool flushing work from the InnoDB master thread.
- **Incompatible Change:** The AES_ENCRYPT() and AES_DECRYPT() functions now permit control of the block encryption mode and take an optional initialization vector argument:

  - The new block_encryption_mode system variable controls the mode for block-based encryption algorithms. Its default value is aes-128-ecb, which signifies encryption using a key length of 128 bits and ECB mode.

  - An optional *init_vector* argument provides an initialization vector for encryption modes that require it:

    ```
    AES_ENCRYPT(str,key_str[,init_vector])
    AES_DECRYPT(crypt_str,key_str[,init_vector])
    ```

  - A random string of bytes to use for the initialization vector can be produced by calling the new RANDOM_BYTES() function.

  For more information, see Encryption and Compression Functions.

  These changes make statements that use AES_ENCRYPT() or AES_DECRYPT() unsafe for statement-based replication and they cannot be stored in the query cache. Queries that use RANDOM_BYTES() are unsafe for statement-based replication and cannot be stored in the query cache.

- **InnoDB:** InnoDB now supports the Transportable Tablespace feature for partitioned InnoDB tables and individual InnoDB table partitions. This enhancement eases backup procedures for partitioned tables and enables copying of partitioned tables and individual table partitions between MySQL instances. For additional information, see Copying Tablespaces to Another Server (Transportable Tablespaces). (Bug #18121824, Bug #70196, Bug #18304194, Bug #71784)

- **InnoDB:** Parameters used to identify an InnoDB pages are replaced by two new classes, and fold value and zip_size value calculations have been optimized. These changes simplify code by reducing the number of page identifier parameters passed to functions. (Bug #18073495)

- **InnoDB:** The Fusion-io Non-Volatile Memory (NVM) file system on Linux provides atomic write capability, which makes the InnoDB doublewrite buffer redundant. For system tablespace files (ibdata files) located on Fusion-io devices that support atomic writes, the InnoDB doublewrite buffer is automatically disabled. (Bug #18069105)

- **InnoDB:** Reverse index leaf page scan has been optimized. btr_pcur_restore_position_func() can now perform optimistic restoration for reverse cursors, which reduces block->mutex contention on the root page, especially for concurrent reverse scans. (Bug #17666170)

- **InnoDB:** A new system variable, innodb_log_write_ahead_size, allows you to configure the write-ahead block size for redo logs to a value that matches the operating system or file system cache block size in order to avoid “read-on-write” for redo log writes. Read-on-write occurs when redo log blocks are not entirely cached to the operating system or file system due to a mismatch between write-ahead block size for redo logs and operating system or file system cache block size. Avoiding read-on-write improves throughput stability for redo log writes. (Bug #17571371)

- **InnoDB:** MySQL now supports rebuilding regular and partitioned InnoDB tables using online DDL (ALGORITHM=INPLACE) for the following operations:

  - OPTIMIZE TABLE

  - ALTER TABLE ... FORCE

  - ALTER TABLE ... ENGINE=INNODB (when run on an InnoDB table)

    Online DDL support reduces table rebuild time and permits concurrent DML, which helps reduce user application downtime. For additional information, see Overview of Online DDL.

  (Bug #13975225)

- **InnoDB:** New global configuration parameters, innodb_status_output and innodb_status_output_locks, allow you to dynamically enable and disable the standard InnoDB Monitor and InnoDB Lock Monitor for periodic output. Previously, enabling and disabling these monitors for periodic output required creating and dropping specially named tables (innodb_monitor and innodb_lock_monitor). For additional information, see InnoDB Monitors.

- **InnoDB:** The InnoDB Tablespace Monitor and InnoDB Table Monitor are removed in MySQL 5.7.4. For the Tablespace Monitor, equivalent functionality will be introduced before the GA release of MySQL 5.7. For the Table Monitor, equivalent information can be obtained from InnoDB INFORMATION_SCHEMA tables.

- **InnoDB:** MySQL now includes a gb18030 character set that supports the China National Standard GB18030 character set. For more information about MySQL character set support, see Character Set Support.

- **Replication:** The binary log dump thread has been optimized by removing unnecessary reallocation of the send buffer. Previously, memory was allocated then freed for every event sent to the slave, even when this was not strictly necessary. Following this optimization, the MySQL Server can make better use of hardware resources by having the dump thread employ adaptative memory allocation, which can also result in less CPU usage. (Bug #11747349, Bug #31932, Bug #11752288, Bug #43426, Bug #13727951)

- **Replication:** Implemented separate threads for sending and receiving semisynchronous replication acknowledgement signals, so that event streams and ACK streams can be sent and received simultaneously. This should reduce many common delays and thus improve performance with semisynchronous replication in a number of settings.

- **Replication:** It is now possible in many cases to execute CHANGE MASTER TO without first issuing STOP SLAVE. This capability is added by implementing the following changes in the behavior of the CHANGE MASTER TO statement, which now depends only on whether the slave SQL thread or slave I/O thread is stopped, as described here:

  - If the SQL thread is stopped, you can execute CHANGE MASTER TO using the RELAY_LOG_FILE, RELAY_LOG_POS, and MASTER_DELAY options, even if the slave I/O thread is running. No other options may be used with this statement when the I/O thread is running.

  - If the I/O thread is stopped, you can execute CHANGE MASTER TO using any of the options for this statement *except* RELAY_LOG_FILE, RELAY_LOG_POS, or MASTER_DELAY, even when the SQL thread is running. These three options cannot be used when the I/O thread is running.

  - Both the SQL thread and the I/O thread must be stopped before issuing CHANGE MASTER TO with MASTER_AUTO_POSITION = 1.

  If you are using statement-based replication and temporary tables, it is possible for a CHANGE MASTER TO statement following a STOP SLAVE statement to leave behind temporary tables on the slave. As part of this set of improvements, a warning is now issued whenever this occurs. You can avoid this in such cases by making sure that Slave_open_temp_tables is equal to 0 prior to executing these statements.

  For more information, see CHANGE MASTER TO Syntax, and Switching Masters During Failover.

- Performance Schema performance was improved in the following ways:

  - When a thread connects, reset of all per-thread statistics is now delayed until a statistic is actually collected. This lazy initialization benefits workloads with very short-lived sessions, for which instrumentation is disabled.

  - When a thread disconnects, the per-thread statistics are aggregated to a parent only for statistics that actually collected data. This optimization benefits workloads with very short-lived sessions, for which instrumentation is disabled.

  - For statement instrumentation, reset of an individual `EVENT_NAME` statistic is also now delayed until a statistic is actually collected. This benefits all workloads that contain only a few types of statements (`SELECT`, `INSERT`, `UPDATE`, `DELETE`, and so forth) from the very large set statements supported in MySQL. Only statements for event names actually executed are aggregated on disconnect.

  - The memory footprint of internal memory buffers is reduced, by removing some attributes reserved for future use, that were in fact not used. In particular, statistics for mutexes, rwlocks and conditions now need less memory.

  (Bug #18324285)

- On Windows, `NOMINMAX` is set using the `ADD_DEFINITIONS()` `CMake` macro rather than in `config.h.cmake` so that it is set even without including `my_config.h`. (Bug #18192896)

- On Solaris, `mysql_config --libs` now includes `-R/*path/to/library*` so that libraries can be found at runtime. (Bug #18235669)

- `CMake` support for compiling MySQL with `gcc` on Solaris was improved. Binary distributions for Solaris now are built using `gcc` rather than Sun Studio, to enable compilation of code not handled by Sun Studio.

  A consequence of this change is that on Solaris, `mysql_config` no longer provides arguments for linking with the embedded library, since this is now built using `gcc` instead of Sun Studio. To get linking arguments for the embedded library, use the alternative script `mysql_server_config` instead. (Bug #18146422, Bug #17826757)

- The `CHECK_FUNCTION_REPLACEMENT()` `CMake` macro was removed from `Windows.cmake` and replacement functions are set explicitly instead since the result of the check was already hard coded in `WindowsCache.cmake`. (Bug #18116661)

- MySQL now compiles using Clang 3.4. (Bug #18047020)

- In MySQL 5.7.1, the MySQL test suite `mysql-test-run.sh` program was modified to start the server with `InnoDB` rather than `MyISAM` as the default storage engine. Test cases in the MySQL test suite that require `MyISAM` now have been marked as such to make it clear that they do not run using `InnoDB`. (Bug #17902011)

- Performance Schema instrumentation was added to capture GTIDs for transaction events. (Bug #17799147)

- Performance Schema overhead was reduced for the `pfs_lock` implementation and the uses of atomic operations in general. (Bug #17766582)

- `CMake` now aborts the configuration process on Windows if a Visual Studio version earlier than 2010 is used. (Bug #17730320)

- A new `CMake` option, `WITH_MSAN`, permits enabling MemorySanitizer for compilers that support it. (Bug #17632319)

- Previously, `ALTER TABLE` in MySQL 5.6 could alter a table such that the result had temporal columns in both 5.5 and 5.6 format. Now `ALTER TABLE` upgrades old temporal columns to 5.6 format for `ADD COLUMN`, `CHANGE COLUMN`, `MODIFY COLUMN`, `ADD INDEX`, and `FORCE` operations. This conversion cannot be done using the INPLACE algorithm, so specifying ALGORITHM=INPLACE in these cases results in an error. (Bug #17246318)

- Overhead was reduced for filesort comparison operations. (Bug #14635144)
- Overhead was reduced for metadata lock acquisition for DML statements. (Bug #58627, Bug #11765641)
- CMake now supports a -DTMPDIR=*dir_name* option to specify the default tmpdir value. If unspecified, the value defaults to P_tmpdir in <stdio.h>. Thanks to Honza Horak for the patch. (Bug #68338, Bug #16316074)
- Based on community feedback, the default value of 10 for the eq_range_index_dive_limit system variable has proven to be too low. The default has been raised to 200. (Bug #70586, Bug #17587952)
- Code instrumented with Valgrind did not preallocate memory in alloc_root(), to help find bugs. This behavior is now also enabled if ASAN (address sanitizer) is used. (Bug #44582, Bug #11753184)
- mysql_install_db provides a more informative diagnostic message when required Perl modules are missing. (Bug #69844, Bug #18187451)
- In MySQL 5.7.1, the MySQL test suite mysql-test-run.sh program was modified to start the server with InnoDB rather than MyISAM as the default storage engine and several cases in the MySQL test suite were modified to include a force_default_myisam.inc file. For such test cases that are intended for storage engines other than InnoDB or MyISAM, that line has been removed.
- The metadata locking subsystem implementation now uses lock-free hashing rather than a hash protected by a mutex. An implication of this change is that the metadata_locks_cache_size and metadata_locks_hash_instances system variables no longer have any effect, so they are deprecated and will be removed in a future MySQL release.
- The deprecated IGNORE clause for ALTER TABLE has been removed and its use produces an error.
- The metadata locking subsystem now uses a lock-free algorithm for acquiring and releasing locks typical for DML statements. This gives better performance/scalability on multi-core machines in workloads involving lots of small read-only transactions.

Bugs Fixed

- **Important Change; InnoDB; Partitioning:** The FLUSH TABLES statement's FOR EXPORT option is now supported for partitioned InnoDB tables. (Bug #16943907)
- **InnoDB; Replication:** Using the InnoDB memcached plugin (see InnoDB Integration with memcached) with innodb_api_enable_binlog set to 1 caused the server to leak memory. (Bug #70757, Bug #17675622)
- **InnoDB:** The server would halt with an assertion in lock_rec_has_to_wait_in_queue(lock) due to a locking-related issue and a transaction being prematurely removed from trx_sys->rw_trx_set. (Bug #173209771)
- **InnoDB:** trx_undo_truncate_start would write to the redo log for temporary undo log segments, resulting in a purge thread assertion. (Bug #18252937)
- **InnoDB:** A !sync_check_iterate(check) assertion occurred in fts_create_doc_id() during an INSERT operation. (Bug #18253210)
- **InnoDB:** The GCC 4.4 Compiler would emit a bogus warnings about InnoDB parsing functions, indicating that output is uninitialized. (Bug #18192536)

- **InnoDB:** log_mutex_exit should be called before log_buffer_extend when the log buffer mutex is held. (Bug #18202904)
- **InnoDB:** InnoDB would write to the redo log for some operations on temporary tablespaces. (Bug #18226934)
- **InnoDB:** Doublewrite buffer error messages referenced page numbers in an inconsistent manner. (Bug #18242594)
- **InnoDB:** The truncate_t::drop_indexes and truncate_t::create_indexes methods would disable redo logging when modifying the system tablespace. (Bug #18236000)
- **InnoDB:** Unused parameters and variables along with disabled functionality has been removed from redo log code. This patch also includes redo log code improvements including test enablement, addition of Valgrind instrumentation, and minor code refactoring. (Bug #18251675)
- **InnoDB:** An invalid memmove in fts_query_fetch_document would cause a serious error. (Bug #18229433)
- **InnoDB:** For full-text queries, a failure to check that num_token is less than max_proximity_item could result in an assertion. (Bug #18233051)
- **InnoDB:** InnoDB would perform unnecessary redo log writes and flushing for temporary tablespaces. (Bug #18236692)
- **InnoDB:** The user defined type, xid_t, was defined multiple times. (Bug #18251254)
- **InnoDB:** To simplify code and reduce memory usage, InnoDB redo log scan records for file-level operations, previously stored in a hash table, are now processed immediately. (Bug #18180875)
- **InnoDB:** The loader in some Solaris versions would refuse to start a GCC-compiled binary (such as the mysqld binary) that uses the PAUSE assembler instruction. (Bug #18122171)
- **InnoDB:** Temporary tablespace file size did not match the file size specified by --innodb-temp-data-file-path due to an error in file size allocation logic. (Bug #18141070)
- **InnoDB:** innodb_ft_result_cache_limit now has a hardcoded maximum value of 4294967295 bytes or (2**32 -1). The maximum value was previously defined as the maximum value of ulong. (Bug #18180057, Bug #71554)
- **InnoDB:** InnoDB would fail to restore a corrupt first page of a system tablespace data file from the doublewrite buffer, resulting in a startup failure. (Bug #18144349, Bug #18058884)
- **InnoDB:** An UPDATE resulted in a memory access error in lock_rec_other_trx_holds_expl. The transaction list (trx_sys->rw_trx_list) was traversed without acquiring the transaction subsystem mutex (trx_sys->mutex). (Bug #18161853)
- **InnoDB:** TRUNCATE TABLE on Windows would report multiple 'chsize' returned OS error 71 errors. (Bug #18176071, Bug #71173)
- **InnoDB:** A regression introduced by Bug #14329288 would result in a performance degradation when a compressed table does not fit into memory. (Bug #18124788, Bug #71436)
- **InnoDB:** In the case of a corrupted clustered index on a temporary table, the server would crash on an INSERT instead of returning an error. In the case of a corrupted clustered index on a normal table, an error was not returned and the INSERT would succeed. (Bug #18064548)
- **InnoDB:** Specifying an alternate directory for InnoDB tables using datadir and then moving .ibd files to the default MySQL data directory would result in a serious error when attempting a DROP TABLE operation. (Bug #18063570)

- **InnoDB:** The `UNIV_LOG_DEBUG` debug flag, which is no longer fully functional, has been removed. (Bug #18080537)
- **InnoDB:** A compilation error (`unable to find string literal operator`) was returned when building in `c++11` mode. (Bug #18082139)
- **InnoDB:** `INFORMATION_SCHEMA.INNODB_TRX` contained a bogus transaction ID that did not match transaction ID values printed elsewhere. The method used to retrieve transaction IDs was inconsistent. (Bug #18111007)
- **InnoDB:** The maximum value for `innodb_thread_sleep_delay` is now 1000000 microseconds. The previous maximum value (4294967295 microseconds on 32-bit and 18446744073709551615 microseconds on 64-bit) was unnecessarily large. Because the maximum value of `innodb_thread_sleep_delay` is limited by the value set for `innodb_adaptive_max_sleep_delay` (when set to a non-zero value), the maximum value for `innodb_thread_sleep_delay` is now the same as the maximum value for `innodb_adaptive_max_sleep_delay`. (Bug #18117322)
- **InnoDB:** The `fil_node_create` function would perform an unnecessary hash table lookup. (Bug #18116588)
- **InnoDB:** When starting the server, unnecessary “checking space” log messages would be printed when processing the doublewrite buffer. (Bug #18101380)
- **InnoDB:** A full-text tokenizer thread would terminate with an incorrect error message. (Bug #18021306)
- **InnoDB:** Attempting to uninstall the `InnoDB` `memcached` plugin while the `InnoDB` `memcached` plugin is still initializing would kill the `InnoDB` `memcached` daemon thread. Uninstall should wait until initialization is complete. (Bug #18038948)
- **InnoDB:** In debug builds, creating a unique index on a binary column, with input data containing duplicate keys, would cause an assertion. (Bug #18010711)
- **InnoDB:** Logging functions were not used consistently. The bug fix replaces occurrences of `fprintf(stderr, message)` and `fputs()` with `ib_logf()`. Also, because `ib_logf()` emits a timestamp with each message, the bug fix removes unnecessary occurrences of `ut_print_timestamp()`. (Bug #17935793, Bug #17534737)
- **InnoDB:** A boolean mode full-text search query would result in a memory access violation during parsing. (Bug #17978763)
- **InnoDB:** The `srv_monitor_thread` would crash in the `lock_print_info_summary()` function due to a race condition between the `srv_monitor_thread` and purge coordinator thread. (Bug #17980590, Bug #70430)
- **InnoDB:** On Windows, a regression introduced in 5.7.3 would allow log writes during sync operations, which should not be allowed due to an issue in some Windows environments. (Bug #17824101)
- **InnoDB:** Due to a parser error, full-text search queries that include a sub-expression could return the wrong result. (Bug #17840768)
- **InnoDB:** The `innochecksum` tool did not use a Windows-specific API to retrieve file size information, which resulted in an incorrect error message (`Error: ibdata1 cannot be found`) when the MySQL 5.6 `innochecksum` 2GB file size limit was exceeded. `innochecksum` now provides support for files larger than 2GB in both MySQL 5.6 and MySQL 5.7. (Bug #17810862, Bug #70936)
- **InnoDB:** Due to a regression introduced by the fix for Bug#17371537, memory was not allocated for the default memcached engine when using the default memcached engine as the backstore for data instead of `InnoDB`. (Bug #17800829)

- **InnoDB:** If a crash occurred while temporary tables are active, InnoDB would report an invalid error message on restart indicating that a temporary table does not exist in the InnoDB internal data dictionary. (Bug #17779729)
- **InnoDB:** A page allocation for an undo log due would fail with a "table is full" error message instead of an "undo log is full" error message. (Bug #17779822)
- **InnoDB:** ut_free could be called more than once in succession. (Bug #17763472)
- **InnoDB:** An index tree modification could result in a deadlock. (Bug #17754767)
- **InnoDB:** Manipulating a table after discarding its tablespace using ALTER TABLE ... DISCARD TABLESPACE could result in a serious error. (Bug #17700280)
- **InnoDB:** A race condition in DebugPolicy::enter() would cause a segmentation fault in sync_array_cell_print. (Bug #17713784)
- **InnoDB:** Persistent optimizer statistics would cause stalls due to latch contention. (Bug #17699331, Bug #70768)
- **InnoDB:** MATCH() ... AGAINST queries that use a long string as an argument for AGAINST() could result in an error when run on an InnoDB table with a full-text search index. (Bug #17640261)
- **InnoDB:** An InnoDB full-text search failure would occur due to an "unended" token. The string and string length should be passed for string comparison. (Bug #17659310)
- **InnoDB:** Databases names beginning with a digit or special character would cause a full-text search (FTS) parser error. (Bug #17607687)

  References: See also Bug #17607956.
- **InnoDB:** For debug builds, the table rebuilding variant of online ALTER TABLE, when run on tables with BLOB columns, would cause an assertion in the row_log_table_apply_update function. For normal builds, a DB_PRODUCTION error would be returned. (Bug #17661919)
- **InnoDB:** Under certain conditions, a regression introduced by the fix for Bug #11758237 would cause an assertion error when INSERT ... ON DUPLICATE KEY UPDATE or REPLACE statements encounter a DB_DUPLICATE_KEY error. (Bug #17604730)
- **InnoDB:** In debug builds, a merge insert buffer during a page read would cause a memory access violation. (Bug #17561188)
- **InnoDB:** The patch for Bug #16852278, which simplifies and optimizes comparison functions in InnoDB, caused a query performance regression. (Bug #17543588)
- **InnoDB:** Truncating a memcached InnoDB table while memcached is performing DML operations would result in a serious error. (Bug #17468031)
- **InnoDB:** In sync0rw.ic, rw_lock_x_lock_func_nowait would needlessly call os_thread_get_curr_id. (Bug #17509710, Bug #70417)
- **InnoDB:** Attempting to rename a table to a missing database would result in a serious error. (Bug #17447500)
- **InnoDB:** If a tablespace data file path is updated in a .isl file and then a crash recovery is performed, the updated tablespace data file path is read from the .isl file but the SYS_DATAFILES table would not be not updated. The SYS_DATAFILES table is now updated with the new data file path after crash recovery. (Bug #17448389)

- **InnoDB:** The server could fail to restart if a crash occurred immediately following a RENAME TABLE in an ALTER TABLE, RENAME TABLE sequence. (Bug #17463290)

- **InnoDB:** If the first page (page 0) of file-per-table tablespace data file was corrupt, recovery would be halted even though the doublewrite buffer contained a clean copy of the page. (Bug #17335427, Bug #70087)

- **InnoDB:** The InnoDB memcached Readme file (README-innodb_memcached) incorrectly stated that libevent 1.6.0 is linked statically into daemon memcached. The bundled version of libevent is 1.4.12, not 1.6.0. (Bug #17324419, Bug #70034)

- **InnoDB:** Attempting to reset a replication slave while innodb_force_recovery is greater than 0 would return a cryptic error message: ERROR(1030) HY000: Got error -1 from storage engine. The error message has been changed to: ERROR HY000: Operation not allowed when innodb_force_recovery > 0. Replication options such as --relay-log-info-repository=TABLE and --master-info-repository=TABLE store information in tables in InnoDB. When innodb_force_recovery is greater than 0, replication tables cannot be updated which may cause replication administration commands to fail. (Bug #17287443, Bug #69907)

- **InnoDB:** When creating a table there are a minimum of three separate inserts on the mysql.innodb_index_stats table. To improve CREATE TABLE performance, there is now a single COMMIT operation instead of one for each insert. (Bug #17323202, Bug #70063)

- **InnoDB:** The ALTER TABLE INPLACE algorithm would fail to decrease the auto-increment value. (Bug #17250787, Bug #69882)

- **InnoDB:** Comments in btr0cur.cc incorrectly stated that btr_cur_pessimistic_update() and btr_cur_optimistic_update() would accept a NULL value. (Bug #17231743, Bug #69847)

- **InnoDB:** dict_table_schema_check would call dtype_sql_name needlessly. (Bug #17193801, Bug #69802)

- **InnoDB:** The function os_file_get_status would not work with raw devices. (Bug #17023438, Bug #69424)

- **InnoDB:** fil_check_first_page() failed to check if fsp_flags_get_zip_size() returned a valid value, which resulted in a segmentation fault when starting mysqld. (Bug #17033182)

- **InnoDB:** The lock_rec_other_has_expl_req function in lock0lock.cc would perform unnecessary work. (Bug #17016214, Bug #69576)

- **InnoDB:** Valgrind would report uninitialized values while running a rollback debug test. The Valgrind warnings should only appear in Valgrind-instrumented builds. (Bug #16969876)

- **InnoDB:** During crash recovery, an incorrect transaction active time would result in rolling back an uncommitted transaction. (Bug #16936961, Bug #69438)

- **InnoDB:** Heap block debugging information (file_name, lineno), used for logging diagnostics, would appear in release builds. This information should only appear in debug builds. (Bug #16924719, Bug #69422)

- **InnoDB:** Renaming a column while also adding or dropping columns in the same ALTER TABLE operation would cause an error. (Bug #16864981)

- **InnoDB:** An online ALTER TABLE operation would consume more memory than expected. During an online ALTER TABLE operation, an online log buffer containing a head and tail buffer is created for each index that is created or rebuilt. The tail buffer is the writer context and is only required for concurrent write operations on an index while the ALTER TABLE operation is in progress. The head buffer is the reader context and is only required during the log apply phase. To reduce memory consumption, the tail buffer is now allocated when the first DML statement is run on the index, and the head buffer is only allocated in the log apply phase and freed afterwards. (Bug #16868967, Bug #69325, Bug #17911720)

- **InnoDB:** A type name (`srv_shutdown_state`) was the same as a variable name. The `srv_shutdown_state` type name has been changed to `srv_shutdown_t`. (Bug #16735398)
- **InnoDB:** On Windows, the full-text search (FTS) object ID was not in the expected hexadecimal format. (Bug #16559254)

  References: See also Bug #16559119.
- **InnoDB:** The `buf_buddy_relocate` function would perform an unnecessary hash lookup. (Bug #16596057)
- **InnoDB:** Server shutdown would result in a hang with the following message written to the error log: "`[NOTE] InnoDB: Waiting for purge thread to be suspended`." (Bug #16495065)
- **InnoDB:** `InnoDB` would fail to start when `innodb_data_file_path` specified the data file size in kilobytes by appending `K` to the size value. (Bug #16287752)
- **InnoDB:** Fetching and releasing pages from the buffer pool and tracking the page state are expensive and complex operations. Prior to the bug fix, these operations were performed using a page mutex. Using a page mutex to track several things is expensive and does not scale well. The bug fix separates fetch and release tracking (in-use state) of a page from page I/O state tracking. Fetch and release is now tracked using atomics where available.

  For portability, a new `CMake` build option, `INNODB_PAGE_ATOMIC_REF_COUNT` (default `ON`), can be used to disable atomic page reference counting on platforms where atomics support is not available. When atomic page reference counting is enabled (default), "`[Note] InnoDB: Using atomics to ref count buffer pool pages`" is printed to the error log at server startup. If atomic page reference counting is disabled, "`[Note] InnoDB: Using mutexes to ref count buffer pool pages`" is printed instead. (Bug #16249481, Bug #68079)
- **InnoDB:** An insert buffer merge would cause an assertion error due to incorrectly handled ownership information for externally stored BLOBs.

  ```
  InnoDB: Assertion failure in thread thread_num in file ibuf0ibuf.cc line 4080
  InnoDB: Failing assertion: rec_get_deleted_flag(rec, page_is_comp(page))
  ```

  (Bug #14668683)
- **InnoDB:** Decreasing the `auto_increment_increment` value would have no affect on the next auto-increment value. (Bug #14049391, Bug #65225)
- **InnoDB:** Table renaming errors would appear in the `LATEST FOREIGN KEY ERROR` section of the `SHOW ENGINE INNODB STATUS` output. (Bug #12762390, Bug #61746)
- **InnoDB:** The page latching algorithm for b-trees would lock sibling leaf pages, prolonging dictionary locks. The bug fix implements prefectching of sibling leaf pages to reduce index lock holding time. (Bug #12734249, Bug #61736)
- **InnoDB:** An `INSERT ...ON DUPLICATE KEY UPDATE` statement run on a table with multiple unique indexes would sometimes cause events to be incorrectly written to the binary log. (Bug #11758237, Bug #50413)
- **InnoDB:** `BUF_READ_AHEAD_AREA` would frequently call `ut_2_power_up` for workloads with a high I/O rate. The calculation is now performed once and the result is stored in the `buf_pool_t` structure. (Bug #11762242, Bug #54814)

- **InnoDB:** UNIV_SYNC_DEBUG, which was disabled in univ.i with the fix for Bug#16720368, is now enabled. (Bug #69617, Bug #17033591)

- **Partitioning:** Queries using the index_merge optimization (see Index Merge Optimization) could return invalid results when run against tables that were partitioned by HASH. (Bug #17588348, Bug #70588)

  References: See also Bug #16862316, Bug #17648468, Bug #18167648.

- **Partitioning:** When no partition had returned a row since the last **HA_ERR_KEY_NOT_FOUND** error, the use of uninitialized memory in the priority queue used for returning rows in sorted order could lead to a crash of the server. (Bug #17401628)

- **Replication:** When running the server with --gtid-mode=ON, STOP SLAVE followed by START SLAVE resulted in a mismatch between the information provided by INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO and the Slave_open_temp_tables status variable: the INNODB_TEMP_TABLE_INFO table showed that no temporary tables existed, but Slave_open_temp_tables had a nonzero value. (Bug #18216612)

- **Replication:** Attempting to use semisynchronous replication concurrently with SSH connections caused the server to fail. (Bug #18219471)

- **Replication:** When MASTER_HEARTBEAT_PERIOD was not included in CHANGE MASTER TO, the statement reset Slave_heartbeat_period to its default value and Slave_received_heartbeats to 0. Now the heartbeat period is not changed by CHANGE MASTER TO unless explicitly set using MASTER_HEARTBEAT_PERIOD. In addition, the statement no longer resets Slave_received_heartbeats. (Bug #18185490)

- **Replication:** The MASTER_SSL_CRL and MASTER_SSL_CRLPATH options are not available when using yaSSL; MySQL Replication now sets these to NULL automatically whenever yaSSL is enabled. (Bug #18165937)

- **Replication:** mysqlbinlog did not free up memory used by its event buffer when using the --rewrite-db option. (Bug #18164998)

- **Replication:** After setting MASTER_SSL_CRLPATH using a CHANGE MASTER TO statement, the option value was not displayed properly in the SSL_CRL_PATH column of the Performance Schema replication_connection_configuration table. (Bug #18174719)

- **Replication:** Setting --slave-parallel-workers to 1 or greater and starting the slave caused the slave SQL thread to use but not release memory until the slave was restarted with STOP SLAVE and START SLAVE. (Bug #18001777, Bug #71197)

- **Replication:** When a slave was configured with replication filters and --log-warnings=2, every statement which was filtered caused an entry to be written in the error log. For busy servers which generated many statements to be filtered, the result was that the error log could quickly grow to many gigabytes in size. Now a throttle is used for such errors, so that an error message is printed only once in a given interval, saying that this particular error occurred a specific number of times during that interval. (Bug #17986385)

- **Replication:** When the master and the slave both had gtid_mode=OFF set initially, and the slave SQL thread was stopped while there remained anonymous transactions in the relay log, if the slave was then restarted with gtid_mode=ON, then the slave assigned GTIDs such transactions. This could cause problems when the slave was later promoted to a master, as the transactions would be played again on the promoted master's slaves, leading quickly to inconsistencies on those slaves. (Bug #17813449)

  References: See also Bug #17827018.

- **Replication:** When the master and the slave both had gtid_mode=ON set initially, and the slave SQL thread was stopped while there remained GTID transactions in the relay log, if the slave was then restarted with gtid_mode=OFF, then the slave SQL thread executed any anonymous transaction it encountered without writing its GTID to the binary log, with the result that the GTID was lost. This could cause problems when the slave was later promoted to a master, as the transaction would be played again on the promoted master's slaves, leading quickly to inconsistencies on those slaves. (Bug #17827018)

  References: See also Bug #17813449.

- **Replication:** When the binary log I/O cache grew to exactly 32768 bytes and the current transaction was preceded by a transaction whose size was greater than 32768 bytes, events could be corrupted when written into the binary log. (Bug #17842137)

- **Replication:** Creating and dropping large numbers of temporary tables could lead to increased memory consumption. (Bug #17806014)

- **Replication:** SHOW SLAVE STATUS used incorrect values when reporting MASTER_SSL_CRL and MASTER_SSL_CRLPATH. (Bug #17772911, Bug #70866)

  References: This bug was introduced by Bug #11747191.

- **Replication:** Binary log events could be sent to slaves before they were flushed to disk on the master, even when sync_binlog was set to 1. This could lead to either of those of the following two issues when the master was restarted following a crash of the operating system:

  - Replication cannot continue because one or more slaves are requesting replicate events that do not exist on the master.

  - Data exists on one or more slaves, but not on the master.

  Such problems are expected on less durable settings (sync_binlog not equal to 1), but it should not happen when sync_binlog is 1. To fix this issue, a lock (LOCK_log) is now held during synchronization, and is released only after the binary events are actually written to disk. (Bug #17632285, Bug #70669)

- **Replication:** mysqlbinlog --verbose failed when it encountered a corrupt row event in the binary log. Such a row event could also cause the slave to fail. (Bug #17632978)

  References: See also Bug #16960133.

- **Replication:** When log_warnings is greater than 1, the master prints binary log dump thread information—containing the slave server ID, binary log file name, and binary log position—in mysqld.1.err. A slave server ID greater than 2 billion was printed with a negative value in such cases. (Bug #17641586, Bug #70685)

- **Replication:** When running the slave with --slave-parallel-workers at 1 or greater, setting --slave-skip-errors=all caused the error log to be filled with with instances of the warning Slave SQL: Could not execute Query event. Detailed error: ;, Error_code: 0. (Bug #17581990, Bug #68429)

  References: See also Bug #17986385.

- **Replication:** When semi-synchronous replication was configured on an independent server with no slaves and rpl_semi_sync_master_wait_no_slave was set to OFF, the master still waited for an ACK from the slave. When rpl_semi_sync_master_wait_no_slave is set to OFF, the master should revert to normal replication when the number of slaves reaches zero during the specified timeout period. Now in such cases the server checks whether semi-synchronous replication is switched on, and, if so, goes on to check whether any slaves are connected. If none are connected, semi-synchronous replication is disabled until such time that the user sets the value of `rpl_semi_sync_master_wait_no_slave` to `ON`. (Bug #17510411, Bug #70360)

- **Replication:** A number of possible state messages used as values for the `PROCESSLIST_STATE` column of the Performance Schema `threads` table were longer than the width of the column (64 characters).

  The long state messages have now been rewritten, and shortened accordingly. This fix applies in MySQL 5.7 and later. (Bug #17319380)

- **Replication:** `Seconds_Behind_Master` in the output of `SHOW SLAVE STATUS` could under some conditions be reported as 0 when it should have had a value greater than zero. (Bug #17233214)

  References: See also Bug #16579028.

- **Replication:** The server did not handle correctly the insertion of a row larger than 4 GB when using row-based replication. (Bug #17081415)

- **Replication:** When using row-based replication, an additional auto-increment column on the slave version of a table was not updated correctly; a zero was inserted instead. (Bug #17066269, Bug #69680)

- **Replication:** Statements involving the Performance Schema tables should not be written to the binary log, because the content of these tables is applicable only to a given MySQL Server instance, and may differ greatly between different servers in a replication topology. The database administrator should be able to configure (`INSERT`, `UPDATE`, or `DELETE`) or flush (`TRUNCATE TABLE`) performance schema tables on a single server without affecting others. However, when replicating from a MySQL 5.5 master to a MySQL 5.5 or later slave, warnings about unsafe statements updating Performance Schema tables were elevated to errors. For MySQL 5.6 and later slaves, this prevented the simultaneous use of `performance_schema` and GTIDs (see Replication with Global Transaction Identifiers).

  This fix causes all updates on tables in the `performance_schema` database to be filtered on the master and not replicated, regardless of the type of logging that is in effect. Prior to this fix, statements using were handled by being marked as unsafe for replication, which caused warnings during execution; the statements were nonetheless written to the binary log, regardless of the logging format in effect.

  Existing replication behavior for tables in the `INFORMATION_SCHEMA` database is not changed by this fix.

  For more information, see MySQL Performance Schema. (Bug #16814264)

  References: See also Bug #14741537, Bug #18259193.

- **Replication:** Invalid event offsets in the binary log were not always handled correctly, which could lead to replication failure. (Bug #16736412, Bug #69087)

- **Replication:** Semisynchronous replication became very slow if there were many dump threads (such as from `mysqlbinlog` or slave I/O connections) working at the same time. It was also found that semisynchronous master plugin functions were called even when the dump connections did not support semisynchronous replication, which led to locking of the plugin lock as well as wasting time on necessary code.

  After this fix, non-semisynchronous dump threads no longer call semisynchronous master functions to observe binary events. (Bug #70218, Bug #17434690)

- **Replication:** The semisynchronous replication plugin was called twice for a DDL statement, incrementing Rpl_semi_sync_master_yes_tx by 2 instead of 1 each time such a statement was executed. (Bug #70410, Bug #17509011)

- During compilation, attempts to create sql_yacc.h could be made from multiple directories simultaneously. (Bug #18319335)

- MySQL distributions for Solaris now include a source tarball for gcc under the share directory, to comply with GPL conditions resulting from inclusion of the C++ runtime library. (Bug #18306484)

- mysql_secure_installation attempted to free memory incorrectly and exited abnormally after a failed attempt to read an option file. (Bug #18255657)

- Compilation failed if MySQL was configured with CFLAGS set to include a -Werror option with an argument. (Bug #18173037)

- A shared libmysqld embedded server library was not built on Linux. (Bug #18123048, Bug #16430656, Bug #68559)

- The default compiler flags are picked up from cmake/build_configurations/compiler_options.cmake. This can be switched off by the CMake -DWITH_DEFAULT_COMPILER_OPTIONS=0 option. However, it could also be switched off for the C or C++ compilers if the CFLAGS or CXXFLAGS environment variables were set.

  Those environment variables now have no such effect. To specify compiler flags, use -DWITH_DEFAULT_COMPILER_OPTIONS=0 option, or the -DCMAKE_C_FLAGS=*flags* and -DCMAKE_C_FLAGS=*flags* options can be used. (Bug #18158812)

- While printing the server version, the mysql client did not check for buffer overflow in a string variable. (Bug #18186103)

- mysql_secure_installation exited if mysql_install_db had been run with the --skip-random-passwords option. (Bug #18181665)

- When the optimizer attempted to use MRR or DS-MRR to read an internally created temporary table, the server could exit or raise an assertion. (Bug #18172819)

- A bug in the range optimizer code that handles index merge could lead to a server exit or missing rows in the result set. (Bug #18136628)

- Building MySQL from source on Windows using Visual Studio 2008 would fail with an identifier not found error due to a regression introduced by the patch for Bug#16249481. (Bug #18057449)

- Repeated rebuilds in the same source tree resulted in libmysqld.a increasing in size each time. (Bug #18082702)

- mysql_config improperly produced nonempty output when invoked with the --libmysqld-libs (or a synonym) if MySQL was configured with the WITHOUT_SERVER option. (Bug #18102839)

- Type casting during LIKE pattern match operations could cause a server exit. (Bug #18114294)

- SHOW GRANTS could be used to view the password hash for a proxied user. Password hash display now requires the SUPER privilege. (Bug #18057514)

- On Microsoft Windows, the rw-lock backup implementation for the my_atomic_* functions was always used. Now, the native Microsoft Windows implementation is used, where available. (Bug #18054042)

- Link failures were fixed on Solaris SPARC and Linux 64-bit platforms. (Bug #18004599)

- When tables are reopened from the table cache and the current thread is not instrumented for the Performance Schema, the server exited attempting to populate `OWNER_THREAD_ID` in the `table_handles` table. (Bug #18047865)
- A memory leak occurred within the Performance Schema during server startup. (Bug #18003651)
- `storage/ndb/CMakeLists.txt` reset the `CMake` cache for some compiler flags for which the result should have been saved. (Bug #17949504)
- Compilation used different warning flags for Clang and GCC, producing different warning output depending on which compiler you use. Warning output is now consistent for the two compilers. (Bug #17959689)
- Building `libevent` was incorrectly dependent on MySQL being configured with the `-DWITH_INNODB_MEMCACHED=1` option. (Bug #17964544)
- On Solaris, configuration failed if no `STL_LIBRARY_NAME` was found. (Bug #17954277)
- The audit log plugin could cause a server exit during log file rotation operations when there were many operations happening for multiple connections. (Bug #17930339)
- The `SUM_SORT_MERGE_PASSES` column value in the Performance Schema `events_statements_summary_by_digest` table was calculated incorrectly. (Bug #17938255)
- During shutdown, a mutex that was still locked could be removed, causing a server exit. (Bug #17959898)
- If the Performance Schema `events_statements_summary_by_digest` table was full when a statement with a new digest was found, the `Performance_schema_digest_lost` status variable was not incremented. (Bug #17935314)
- `FORCE INDEX [FOR ORDER BY] (`*`index_name`*`)` did not work for joins.

  The fix for this bug also also changes the warning created for `EXPLAIN`. Instead of printing only `{IGNORE|USE|FORCE} INDEX` it now also prints `FOR {GROUP BY|ORDER BY|JOIN}` if that was specified in the query. (Bug #17889511)
- Shutdown of open connection threads could fail to occur cleanly during server shutdown. (Bug #17863690)
- `DECIMAL NOT NULL` items could return `NULL` in subqueries. (Bug #17921777)
- `mysql_secure_installation` exited if it connected using SSL and the user had an expired password. (Bug #17880395)
- With the compressed client/server protocol enabled, Performance Schema statement instrumentation could raise an assertion. (Bug #17794846)
- For debug builds, inserts into a multiple-table view could raise an assertion. (Bug #17834434)
- The optimizer could push down a condition when the index did not have the key part present in the condition. (Bug #17814492)
- For debug builds, the `filesort` algorithm could raise a spurious assertion. (Bug #17734642)
- The `mysql_get_option` symbol was missing from `libmysql.dll`. (Bug #17733103)
- Build and execution problems were fixed for builds made with `gcc` 4.8.1 in 32-bit mode on SPARC. (Bug #17740390)

- The server could exit when executing an INSERT ... SELECT with UNION, ROLLUP, and ON DUPLICATE KEY UPDATE with a subquery. (Bug #17727506)

  References: This bug is a regression of Bug #16967281.

- The patch for Bug #16041903 introduced an incorrect DBUG_ASSERT that in debug builds raised a spurious assertion. (Bug #17746721)
- Valgrind errors were produced during row comparator setup. (Bug #17755540)
- In some cases, UNIX_TIMESTAMP() could return NULL when it should return 0. (Bug #17728371)
- An assertion could be raised if a filesort failed to resize its main buffer when record properties changed. (Bug #17757914)
- The resetconnection command for mysql did not report proper errors if the server was down or the user password had expired. (Bug #17772561)
- Compilation failed if MySQL was configured using -DWITH_LIBWRAP=1. (Bug #17738805)
- Contraction information in a collation could be mishandled, resulting in incorrect decisions about whether a character is part of a contraction, and miscalculation of contraction weights. (Bug #17760379)
- The cache used for the Index Merge access method was freed only after successful retrieval of all rows. Interruption or failure of the operation led to a file descriptor leak. (Bug #17708621)
- The optimizer calculated the cost for joined buffer scans incorrectly, evaluating rows filtered out by attached conditions not once, but once per join buffer. (Bug #17694707)
- Incorrect reference counting in the range optimizer module resulted in potential for missing or duplicate rows in the query result set. (Bug #17619119)
- Using the mysqldump --set-gtid-purged option with no value caused mysqldump to crash. (Bug #17650245)
- A race condition between Performance Schema statement event threads led to a server exit. (Bug #17637970)
- For debug builds, an aggregate function in a subquery join condition could raise an assertion. (Bug #17602807)
- If SAFE_MUTEX was enabled (true only for debug builds), THR_LOCK_mutex was used before being initialized. (Bug #17641055, Bug #70639)
- After the fix for Bug #16409270, it was not possible to #include <mysql.h> following #include <windows.h>. (Bug #17514554)
- An addressing error in accessing the join buffer could produce invalid results or a server exit. (Bug #17513341)
- Old clients (older than MySQL 5.5.7) failed to parse authentication data correctly if the server was started with the --default-authentication-plugin=sha256_password option. (Bug #17495562)
- The parser permitted some queries with multiple ORDER BY clauses, which then failed during execution and caused a server exit. (Bug #17473479)
- For debug builds, the server could exit for statements that inserted into a BLOB column declared as NOT NULL using a subquery that retrieved from a BLOB column and included GROUP BY NULL. (Bug #17458917)

- Insufficient cleanup after removal of a `SELECT_LEX` structure caused dereferencing of a NULL pointer and a server exit. (Bug #17458169)
- Within a `CASE` expression, use of a subquery referencing the `VALUES()` function could cause a server exit. (Bug #17458914)

  References: This bug is a regression of Bug #14789787.
- `mysql_config` incorrectly included some flags to generate compiler warning output. (Bug #17400967)
- The parser silently accepted duplicate `ORDER BY` clauses and/or `LIMIT` clauses before `ORDER BY` clauses in subqueries. These caused failures during query execution. Fixing this problem results in some changes in parser behavior. The parser no longer accepts:
  - A `LIMIT` clause before an `ORDER BY` clause
  - A `LIMIT` clause in a parentheses-less `SELECT` statement before a `UNION` keyword
  - An `INTO` clause after a `PROCEDURE ANALYSE()` clause

  (Bug #17426017, Bug #17703542, Bug #17727401)
- `SET PASSWORD` combined with assignment of a variable from a subquery result could raise an assertion. (Bug #17458750)
- On Windows, `mysql_secure_installation` exited if the `root` password was expired. (Bug #17415203)
- For `CASE` expressions involving floating-point numbers, the `max_length` and `decimal` values could be computed incorrectly. The logic for `CASE` was corrected to be the same as for `COALESCE()`, which performs a similar operation. (Bug #17388045)
- If accepting a connection attempt failed due to an out-of-memory error, the server could access a stale thread structure for a previously disconnected connection, resulting in Valgrind errors. (Bug #17398792)
- With semi-join optimization enabled, queries with nested subqueries could cause a server exit due to incorrect resolution of references to columns in the middle query block. (Bug #17398972)
- The SHA256 password authentication algorithm allocated a buffer one byte too short. (Bug #17397073)
- A client crash occurred if `mysql_set_server_option()` or several other C API functions were called before `mysql_real_connect()`. (Bug #17338958)
- In some cases, the optimizer wrote fixed-length temporary `MyISAM` tables to disk rather than variable-length temporary tables. (Bug #17231940)
- Enabling the `validate_password` plugin could result in incorrect password hashes being stored in the `mysql.user` table. (Bug #17065383)
- For debug builds, the second execution of a prepared statement processed using a semi-join could cause a server exit. (Bug #16988465)
- A spurious assertion was raised for queries processed using a semi-join LooseScan optimization that required rows to be returned in order. (Bug #16977389)
- A circular dependency problem involving `sql/sql_builtin.cc` was resolved. (Bug #16877045)
- For accounts authenticated using the `sha256_password` plugin, setting the password after the password had been expired did not clear the password-expired flag. (Bug #16872181)

- During server shutdown, file information was freed before calling `query_logger.cleanup()`, leading to a memory leak. (Bug #16859266)
- For prepared `INSERT INTO ... SELECT` statements, nonexistent column names were not reported during statement preparation, but only later at statement execution. (Bug #16820562)
- Multiple-table updates failed to update under certain conditions. (Bug #16767011)
- Crash recovery of temporary tables used uninitialized memory. (Bug #16754540)
- On Mac OS X 10.7, a race condition involving `vio_shutdown()` and the select-based implementation of `vio_io_wait()` could cause a server exit. (Bug #16354789, Bug #17733393)
- Host names in example URLs used within the source code were replaced by names in the example.com domain, the domain intended by IANA for this purpose. (Bug #15890092)
- On Mac OS X, preloading of client plugins specified with the `LIBMYSQL_PLUGINS` environment variable could fail unless the plugins were located in the hardwired default plugin directory. The C API now checks during plugin preloading for a `LIBMYSQL_PLUGIN_DIR` environment variable which can be set to the path name of the directory in which to look for client plugins. (Bug #13994567, Bug #18110355)
- For `utf8` and `utf8mb4` strings, handler functions unnecessarily called a Unicode conversion function. (Bug #14057034)
- Certain `(... NULL ...) IN (...)` expressions returned `NULL` when they should return 0, such as `SELECT (NULL, 1) IN ((0, 0), (0, 0))`. (Bug #13944462)
- Several `-W` warning flags were turned off for compilation in maintainer mode if MySQL was configured with `-DWITH_INNODB_MEMCACHED=1`. (Bug #13898319)
- The optimizer set up for dynamic range access in some cases where range access cannot be used, resulting in fallback to a table scan. (Bug #13814468)
- Executing `mysqladmin shutdown` for a server running with the thread pool plugin enabled and servicing a large number of concurrent connections caused the server to exit abnormally. (Bug #13788920)
- Calling the `ExtractValue()` function with an invalid XPath expression could in some cases lead to a failure of the server. (Bug #12428404, Bug #61065)
- Use of a nonmulti-byte algorithm for skipping leading spaces in multi-byte strings could cause a server exit. (Bug #12368495)
- With `ONLY_FULL_GROUP_BY` SQL mode enabled, a query that uses `GROUP BY` on a column derived from a subquery in the `FROM` clause failed with a `column isn't in GROUP BY` error, if the query was in a view. (Bug #11923239)
- For the `utf8_bin` collation, `ORDER BY LOWER(`*`col_name`*`)` could produce incorrect ordering. (Bug #69005, Bug #16691598)
- Several issues identified by the Coverity static analysis tool were fixed. Thanks to Honza Horak for the patch. (Bug #70830, Bug #17760511)
- On Windows, the `--local-service` server option did not work, and was not displayed in the `--help` message. (Bug #69637, Bug #17049656)
- The server wrote an excessive number of “Sort aborted” messages to the error log. (Bug #70173, Bug #17372396)

- It was not possible to query a view with an `ORDER BY` clause that referenced an alias in the `SELECT` clause of the view definition, unless all columns in the view were named in the select list.

  To handle this problem, the server now writes a view differently into the `.frm` file that stores the view definition. If you experience view-evaluation errors such as just described, drop and recreate the view so that the `.frm` file contains the updated view representation. (Bug #69678, Bug #17077305)

- Overhead was reduced within critical sections of the `my_fopen()` and `my_register_filename()` `mysys` functions. Thanks to Po-Chun Chang for the patch. (Bug #70848)

- The prototype of the Performance Schema instrumentation API `mysql_cond_timedwait()` call was fixed to be drop-in compatible with `pthread_cond_timedwait()`. This fix affects only implementers of third-party plugins. (Bug #70628, Bug #17702677)

- For debug builds, JSON-format `EXPLAIN` statements for queries that involve semi-join materialization could cause a server exit. (Bug #70014, Bug #17305943)

- For an existing user, `GRANT` with an empty password (`IDENTIFIED BY [PASSWORD] ''`) did not change the password. (Bug #69899, Bug #17256161)

- The `make_atomic_cas_body64` implementation on IA32 with `gcc` but without `gcc` builtins could be miscompiled due to an incorrect constraint. The patch also causes MySQL to use builtin atomics when compiled using `Clang`. (Bug #63451, Bug #17242996)

- The `wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_sync` mutex was not properly instrumented for the Performance Schema. (Bug #70939, Bug #17813333)

- Complex updates of Performance Schema tables involving joins or subqueries failed to update every row. (Bug #70025, Bug #17309657)

- For the path specified with the `--basedir` option, `mysql_plugin` attempted to unlink the path rather than free the memory in which the path was stored. (Bug #69752, Bug #17168602)

- Compilation problems were fixed for errors reported by `Clang` and `gcc` when compiling in C++11 mode. (Bug #66803, Bug #14631159)

- The optimizer could choose `ref` access over `eq_ref` access when cost of a nonunique access was evaluated before cost of a unique index. (Bug #54808, Bug #11762236)

- COUNT(DISTINCT) sometimes produced an incorrect result when the last read row contained a `NULL` value. (Bug #68749, Bug #16539979, Bug #71028, Bug #17867117)

- When run by `root`, `mysqld --help --verbose` exited with a nonzero error code after displaying the help message. (Bug #70058, Bug #17324415)

- MySQL client programs from a Community Edition distribution could not connect using SSL to a MySQL server from an Enterprise Edition. This was due to a difference in certificate handling by yaSSL and OpenSSL (used for Community and Enterprise, respectively). OpenSSL expected a blank certificate to be sent when not all of the `--ssl-ca`, `--ssl-cert`, and `--ssl-key` options were specified, and yaSSL did not do so. To resolve this, yaSSL has been modified to send a blank certificate when an option is missing. (Bug #68788, Bug #16715064)

- `sql_resolver.cc` referred to partitioning code that should have been protected by an `#ifdef`, even when MySQL was configured with `-DWITH_PARTITION_STORAGE_ENGINE=OFF`. (Bug #71010, Bug #17876794)

- An incorrect result could be returned for a query with an `IF()` predicate in the `WHERE` clause combined with `OUTER JOIN` in a subquery that is transformed to a semi-join. (A workaround is to disable semi-join using `SET optimizer_switch='semijoin=off';`) (Bug #70608, Bug #17600176)

- Downloading of the Google Mock library could fail during configuration. This is fixed by requiring CMake 2.8.2 or higher. (Bug #69854, Bug #17231722)
- A full-text search combined with derived tables (subqueries in the FROM clause) caused a server exit.

  Now if a full-text operation depends on a derived table, the server produces an error indicating that a full-text search cannot be done on a materialized table. (Bug #68751, Bug #16539903)
- Some scripts displayed out-of-date information regarding where to report bugs. (Bug #68742, Bug #16530527)
- CMake produced not-useful warnings about INTERFACE_LINK_LIBRARIES policy. (Bug #71089, Bug #17905155, Bug #17894997)
- CMake produced a warning in ssl.cmake due to malformed syntax. (Bug #71094, Bug #17905144)
- Previously, the first stage executed within a statement was stage/sql/init. This collided with a different stage named init and was incompatible with the starting stage for SHOW PROFILE. The first stage executed within a statement is now named stage/sql/starting. (Bug #71201, Bug #17993294)
- Some files in the Performance Schema file_instances table were not being removed because the file-removal operation was not instrumented. (Bug #69782, Bug #17209750)
- mysqldump --single-transaction acquired metadata locks for each dumped table but did not release them until the dump operation finished. Consequently, other DDL operations on a dumped table blocked even after the table itself had been dumped. mysqldump now attempts to release metadata locks earlier. (Bug #71017, Bug #17862905)
- Updating a FEDERATED table with UPDATE... JOIN caused a server exit when the local table contained a single row and that row could be joined to a row in the FEDERATED table. (Bug #68354, Bug #16324629)
- The mysqladmin, mysqlbinlog, mysqlcheck, mysqldump, mysqlimport, mysqlslap, and mysqlshow programs now support a --secure-auth option that prevents sending passwords to the server in old (pre-4.1) format. This option is enabled by default; use --skip-secure-auth to disable it. (Bug #69051, Bug #16723046)
- During configuration, CMake improperly checked for the C++ header file cxxabi.h. (Bug #71268, Bug #18147458)
- On Mac OS X, the libmysqlclient dylib file linked to itself. (Bug #61699, Bug #13890998, Bug #61243, Bug #12590037)
- gcov printed warnings without file names. (Bug #33269, Bug #11747622)
- cmake/configure.pl listed instances of WITH_COMMENT rather than the correct option COMPILATION_COMMENT. (Bug #65834, Bug #14298560)
- FLUSH STATUS cleared a variable that could result a subsequent implicit commit of an XA transaction causing a server exit. (Bug #70854, Bug #17911445)
- A deadlock error occurring during subquery execution could cause an assertion to be raised. (Bug #69969, Bug #17307201)
- The Performance Schema stage/sql/Waiting to get readlock instrument is no longer used and has been removed. (Bug #71298, Bug #18035404)

- A query that creates a temporary table to find distinct values and has a constant value in the projected list could produce incorrect results. (Bug #70657, Bug #17634335)

- `mysqlbinlog` leaked memory in relation to `--rewrite-db` processing. (Bug #71283, Bug #18027692)

- Messages written by the server to the error log for LDML collation definition problems were missing the collation name. (Bug #68144, Bug #16204175)

- Configuring with `-DWITH_DEBUG=1` did not have the same effect as configuring with `-DCMAKE_BUILD_TYPE=Debug`. (Bug #70647, Bug #17632854)

- `mysqlcheck` did not correctly handle table names containing dots. (Bug #68015, Bug #16064833)

- Previously, for `EXPLAIN` output, the rows-examined estimate for Performance Schema tables always displayed as 1000. Now a more accurate estimate is displayed based on sizing parameters used when allocating memory for each table. This results in no change of behavior because Performance Schema tables have no indexes. (Bug #71278, Bug #18024455)

- Optimizer trace output from the range optimizer could include raw binary data and generate unprintable characters. Now binary data is printed in hex format. (Bug #71273, Bug #18023222)

- The `-DWITH_EXAMPLE_STORAGE_ENGINE=1` `CMake` option was ignored but should not have been. If `-DWITH_EXAMPLE_STORAGE_ENGINE=0` is given, the `EXAMPLE` storage engine is built as a plugin. (Bug #70859, Bug #17772560)

  References: See also Bug #18324650.

- For logarithmic functions, if the argument is less than or equal to 0.0E0, response is now as follows: In strict mode, the error “Invalid argument for logarithm” is reported. In non-strict mode, a `NULL` value is returned. (Bug #50507, Bug #11758319)

- Aggregating the results of a subquery in the `FROM` clause could produce incorrect results. (Bug #71244, Bug #18014565)

- Some `BETWEEN` expressions on unsigned values were evaluated using signed arithmetic. Thanks to Xiaobin Lin for the patch. (Bug #70622, Bug #17606942)

- For system variables that take a string value, `SET` statements permitted an unquoted value, but values that contained dots were parsed incorrectly and only part of the value was assigned. For example, `SET GLOBAL slow_query_log_file = my_slow.log` assigned the value `my_slow`. Now such values must be quoted or an error occurs. (Bug #69703, Bug #17075846)

- `mysql_install_db` referred to the obsolete `mysqlbug` script for reporting problems. It now refers to http://bugs.mysql.com/ instead. (Bug #29716, Bug #11746921)

- The deprecated `msql2mysql`, `mysql_convert_table_format`, `mysql_find_rows`, `mysql_fix_extensions`, `mysql_setpermission`, and `mysqlaccess` utilities were removed. (Bug #27482, Bug #69012, Bug #69014, Bug #69015, Bug #69016, Bug #69017, Bug #11746603, Bug #16699248, Bug #16699279, Bug #16699284, Bug #16699317, Bug #18179576)

Changes in MySQL 5.7.3 (2013-12-03, Milestone 13)

> **Note**
>
> This is a milestone release, for use at your own risk. Significant development changes take place in milestone releases and you may encounter compatibility

issues, such as data format changes that require attention in addition to the usual procedure of running mysql_upgrade. For example, you may find it necessary to dump your data with mysqldump before the upgrade and reload it afterward.

Optimizer Notes

- The server no longer uses a temporary table for UNION statements that meet certain qualifications. Instead, it retains from temporary table creation only the data structures necessary to perform result column typecasting. The table is not fully instantiated and no rows are written to or read from it; rows are sent directly to the client. As a result, The result is reduced memory and disk requirements, and smaller delay before the first row is sent to the client because the server need not wait until the last query block is executed. EXPLAIN and optimizer trace output will change: The UNION RESULT query block will not be present because that block is the part that reads from the temporary table.

  The conditions that qualify a UNION for evaluation without a temporary table are:

  - The union is UNION ALL, not UNION or UNION DISTINCT.
  - There is no global ORDER BY clause.
  - The union is not the top-level query block of an {INSERT | REPLACE} ... SELECT ... statement.

  (Bug #50674, Bug #11758470)

- The modified filesort algorithm now includes an additional optimization designed to enable more tuples to fit into the sort buffer: For additional columns of type CHAR or VARCHAR, or any nullable fixed-size data type, the values are packed. For example, without packing, a VARCHAR(255) column value containing only 3 characters takes 255 characters in the sort buffer. With packing, the value requires only 3 characters plus a two-byte length indicator.

  For data containing packable strings shorter than the maximum column length or many NULL values, more records fit into the sort buffer. This improves in-memory sorting of the sort buffer and performance of disk-based merge sorting of the temporary file.

  In edge cases, packing may be disadvantageous: If packable strings are the maximum column length or there are few NULL values, the space required for the length indicators reduces the number of records that fit into the sort buffer and sorting is slower in memory and on disk.

  Packing is not applicable if the filesort uses a priority queue for sorting, as is the case when an ORDER BY ... LIMIT optimization is applied (see Optimizing LIMIT Queries).

  If a filesort is done, optimizer trace output includes a filesort_summary block. For example:

```
"filesort_summary": {
  "rows": 100,
  "examined_rows": 100,
  "number_of_tmp_files": 0,
  "sort_buffer_size": 25192,
  "sort_mode": "<sort_key, packed_additional_fields>"
}
```

  The sort_mode value provides information about the algorithm used and the contents of the sort buffer:

```
<sort_key, rowid>: sort using row pointers
<sort_key, additional_fields>: sort using additional fields
<sort_key, packed_additional_fields>: sort using packed additional fields
```

For additional information about the `filesort` algorithm, see `ORDER BY` Optimization. For information about the optimizer trace, see MySQL Internals: Tracing the Optimizer.

Packaging Notes

- Previously, MySQL Server distributions included the MySQL Reference Manual in Info format (the Docs/mysql.info file). Because the license for the manual restricts redistribution, its inclusion in Community packages caused problems for downstream redistributors, such as those who create Linux distributions. Community distributions of MySQL Server no longer include the mysql.info file, to make the repackaging and redistribution process easier (for example, the source tarball and its checksum can be used directly). This change applies to all source and binary Community packaging formats. Commercial (Enterprise) distributions are unchanged.

  For those who wish to continue using the MySQL Reference Manual in Info format, we have made it available at http://dev.mysql.com/doc/.

Performance Schema Notes

- The Performance Schema now exposes metadata lock information:

  - Locks that have been granted (shows which sessions own which current metadata locks)

  - Locks that have been requested but not yet granted (shows which sessions are waiting for which metadata locks).

  - Lock requests that have been killed by the deadlock detector or timed out and are waiting for the requesting session's lock request to be discarded

  This information enables you to understand metadata lock dependencies between sessions. You can see not only which lock a session is waiting for, but which session currently holds that lock.

  The Performance Schema now also exposes table lock information that shows which table handles the server has open, how they are locked, and by which sessions.

  These specific changes were implemented:

  - The `metadata_locks` and `table_handles` tables list current locks and lock requests for metadata locks and table locks.

  - The `setup_instruments` table now has a `wait/lock/metadata/sql/mdl` instrument for metadata locks. This instrument is disabled by default.

  - The `performance_schema_max_metadata_locks` system variable configures the maximum number of metadata locks tracked in the `metadata_locks` table. For `table_handles`, the size is configured by the existing `performance_schema_max_table_handles` system variable.

  - The `Performance_schema_metadata_lock_lost` status variable indicates the number of times a metadata lock could not be recorded. For `table_handles`, tables that are opened but cannot be instrumented are counted by the existing `Performance_schema_table_handles_lost` status variable.

  For more information, see Performance Schema Lock Tables.

  If you upgrade to this release of MySQL from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `performance_schema` database.

- The Performance Schema now instruments transactions. The information collected includes quantitative and qualitative data including transaction duration, transaction counts, and frequency of various transaction attributes such as isolation level and access modes. This information is collected in tables that contain current and recent transaction events, and is aggregated in summary tables across several dimensions, including user, account, and thread (client connection).

  These new tables store transaction events:

  - `events_transactions_current`: Current transaction events
  - `events_transactions_history`: The most recent transaction events for each thread
  - `events_transactions_history_long`: The most recent transaction events overall

  There are also summary tables that provide aggregated transaction information.

  Within the event hierarchy, wait events nest within stage events, which nest within statement events, which nest within transactions. To reflect this, the `NESTING_EVENT_TYPE` column, in those tables that have it, permits a new value, `TRANSACTION`, in addition to the existing values `STATEMENT`, `STAGE`, and `WAIT`.

  To permit control over configuration of transaction event collection, these changes were made to Performance Schema setup tables:

  - The `setup_instruments` table contains a new instrument named `transaction`. This instrument is disabled by default.
  - The `setup_consumers` table contains new consumer values with names corresponding to the current and recent transaction event table names. These consumers may be used to filter collection of transaction events. Only `events_transactions_current` is enabled by default.
  - The `setup_timers` table contains a new row with a `NAME` value of `transaction` that indicates the unit for transaction event timing. The default unit is `NANOSECOND`.

  For more information, see Performance Schema Transaction Tables, and Transaction Summary Tables.

  If you upgrade to this release of MySQL from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `performance_schema` database.

Security Notes

- **Incompatible Change:** Previously, the `--ssl` option has been treated as advisory: When given, an SSL connection was permitted but not required. Also, several other `--ssl-*xxx*` options implied `--ssl`. Because of this, the option was usually not used explicitly as `--ssl`, but in its negated form as `--ssl=0`, which prevents use of SSL. This was true on both the client and server sides, and true for any synonyms of `--ssl` (`--ssl=1`, `--enable-ssl`) or `--ssl=0` (`--skip-ssl`, `--disable-ssl`).

  Now the meaning of `--ssl` has changed *on the client-side only*. (There are no SSL changes on the server side.)

  When given on the client side as `--ssl` (or a synonym), the option now is prescriptive, not advisory: The client connection must use SSL or the connection attempt fails. In addition, other SSL options no longer imply `--ssl`. This is an incompatible change in the sense that MySQL client commands that use `--ssl` now will fail unless an SSL connection can be established. On the other hand, for a successful connection attempt, the connection is guaranteed to use SSL. Previously, there was no such guarantee.

If other `--ssl-`*`xxx`* options are given in the absence of `--ssl`, the client attempts to connect using SSL. If the server is not configured to permit SSL, the client falls back to an unencrypted connection.

There is no change in the meaning of `--ssl=0` (and its synonyms) to prevent use of SSL and override other SSL options.

A new `MYSQL_OPT_SSL_ENFORCE` option is available for the `mysql_options()` C API function to indicate whether to require the connection to use SSL. If enabled, it has the same effect as specifying `--ssl` on the command line: If an encrypted connection cannot be established, the connection attempt fails.

For more information, see SSL Command Options, and `mysql_options()`.

The `MASTER_SSL=1` option for the `CHANGE MASTER TO` statement has changed as well, analogous to the change in the meaning of `--ssl`. That is, when given, the slave connection to the master must use SSL or the connection attempt fails.

Functionality Added or Changed

- **Performance; InnoDB:** The `log_write_up_to` function, which writes to redo log files up to a certain log sequence number (LSN) and optionally flushes writes to disk, has been refactored to improve performance for workloads with heavy `log_sys::mutex` contention and where `innodb_flush_log_at_trx_commit=2`.

- **Performance:** The `LOCK_thread_count` mutex protected several independent internal server structures and variables, and was a bottleneck, particularly affecting server performance in the circumstance when many clients were connecting and disconnecting at once. This mutex was decomposed into more specific mutexes and atomic operations to alleviate the bottleneck and improve performance.

  As part of this work, the following status variables are no longer visible in the embedded server because for that server they were not updated and were not meaningful: `Aborted_connects`, `Connection_errors_accept`, `Connection_errors_internal`, `Connection_errors_max_connections`, `Connection_errors_peer_addr`, `Connection_errors_select`, `Connection_errors_tcpwrap`.

- **Incompatible Change:** Several statement instruments in the `setup_instruments` table are used by the Performance Schema during the early stages of statement classification before the exact statement type is known. These instruments were renamed to more clearly reflect their “abstract” nature:

Old Instrument Name	New Instrument Name
`statement/com/`	`statement/abstract/new_packet`
`statement/com/Query`	`statement/abstract/Query`
`statement/rpl/relay_log`	`statement/abstract/relay_log`

  In addition, statistics for abstract instruments are no longer collected in the following tables, because no such instrument is ever used as the final classification for a statement:

  ```
  events_statements_summary_by_thread_by_event_name
  events_statements_summary_by_account_by_event_name
  events_statements_summary_by_user_by_event_name
  events_statements_summary_by_host_by_event_name
  events_statements_summary_global_by_event_name
  ```

Applications that refer to the old instrument names must be updated with the new names. For more information about the use of abstract instruments in statement classification, see Performance Schema Statement Event Tables. (Bug #16750433, Bug #17271055)

- **Incompatible Change:** The `EXPLAIN` statement has been changed so that the effects of the `EXTENDED` and `PARTITIONS` keywords are always enabled. `EXTENDED` and `PARTITIONS` are still recognized, but are superfluous and have been deprecated. They will be removed from `EXPLAIN` syntax in a future MySQL release.

  `EXPLAIN` output differs as follows as a result of this change:

  - The filtered and partitions columns appear in `EXPLAIN` output regardless of whether the `EXTENDED` and `PARTITIONS` keywords are specified. This is an incompatible change for applications that expect to identify column information by position rather than by name, and such applications will need adjustment.

  - `SHOW WARNINGS` immediately following `EXPLAIN` shows additional execution plan information regardless of whether the `EXTENDED` keyword is specified. (An additional deprecation warning is included if the statement includes the `EXTENDED` or `PARTITIONS` keyword.)

- **Important Change; InnoDB:** `InnoDB` now supports external full-text parser plugins. In order to support `InnoDB` full-text parser plugins that are called in boolean mode, a new “`position`” member has been added to the `MYSQL_FTPARSER__BOOLEAN_INFO` structure. If you plan to use an existing full-text parser plugin that is called in boolean mode with MySQL 5.7.3 or later, you must add support for the new “`position`” member, which is described in Writing Full-Text Parser Plugins. Altering a `MyISAM` table with a full-text parser plugin to use `InnoDB` is also supported. For additional information about full-text parser plugins, see Full-Text Parser Plugins.

- **InnoDB:** The `InnoDB` `memcached` plugin now supports inserts and reads on mapped `InnoDB` tables that have an `INTEGER` defined as the primary key. (Bug #17315083, Bug #17203937)

- **Replication:** Replication filtering rules can now be set dynamically on the slave using the SQL statement `CHANGE REPLICATION FILTER` introduced in this release. This statement has the same effect as starting the slave `mysqld` with one or more of the options `--replicate-do-db`, `--replicate-ignore-db`, `--replicate-do-table`, `--replicate-ignore-table`, `--replicate-wild-do-table`, `--replicate-wild-ignore-table`, and `--replicate-rewrite-db`.

  For example, issuing the statement `CHANGE REPLICATION FILTER REPLICATE_DO_TABLE = (d1.t2)` is equivalent to starting the slave `mysqld` with `--replicate-do-table='d1.t2'`.

  `CHANGE REPLICATION FILTER` differs from the server options in that, to take effect, the statement requires only that the slave SQL thread be stopped beforehand and restarted afterwards, using `STOP SLAVE SQL_THREAD` and `START SLAVE SQL_THREAD`, respectively.

  This statement leaves any existing replication filtering rules unchanged; to unset all filters of a given type, set the filter to an empty list, as shown in this example:

  ```
  CHANGE REPLICATION FILTER REPLICATE_DO_DB = ();
  ```

  You can list multiple replication filtering rules in the same statement, separated by commas. When multiple instances of the *same* rule are found, only the last instance is used.

  For more information, see `CHANGE REPLICATION FILTER` Syntax; see also How Servers Evaluate Replication Filtering Rules. (Bug #15877941, Bug #11752237, Bug #67362, Bug #43366)

- **Replication:** Previously, with semisynchronous replication enabled, the master waited for a single slave acknowledgment per transaction before proceeding. A new system variable, `rpl_semi_sync_master_wait_for_slave_count`, enables the number of slave acknowledgments required per transaction to be configured. The minimum (and default) value is 1. The maximum is 65,536. Performance is best for small values of this variable.

- Overhead for Performance Schema instrumentation associated with thread creation was reduced. (Bug #17539520)

- The Performance Schema now instruments the read/write lock `Delegate::lock`, which is used for the following classes:

  ```
  Trans_delegate
  Binlog_storage_delegate
  Binlog_transmit_delegate
  Binlog_relay_IO_delegate
  ```

  A different instrument name is used for each subclass, to have distinct statistics for distinct uses. The instruments are visible in the `schema.setup_instruments` table and have these names:

  ```
  wait/synch/rwlock/sql/Trans_delegate::lock
  wait/synch/rwlock/sql/Binlog_storage_delegate::lock
  wait/synch/rwlock/sql/Binlog_transmit_delegate::lock
  wait/synch/rwlock/sql/Binlog_relay_IO_delegate::lock
  ```

  (Bug #17590161, Bug #70577)

- Some dependencies between client-side plugin header files were removed:

  - The `MYSQL_PLUGIN_EXPORT` macro required by plugin declarations is now declared directly in `mysql/client_plugin.h` instead of getting the definition from `mysql/plugin.h`. That macro was the only thing required by client-side plugins and declared in server-side header `mysql/plugin.h`, so including `mysql/client_plugin.h` in an application no longer requires the application to also include `mysql/plugin.h`.

  - `mysql/plugin_trace.h` no longer uses `C_MODE_START` or `C_MODE_END`. Consequently, including `mysql/plugin_trace.h` in an application no longer requires the application to also include `my_global.h`.

  Applications might require `mysql/plugin.h` or `my_global.h` for other reasons, of course. (Bug #17582168)

- It is now possible to enable the Performance Schema but exclude certain parts of the instrumentation. For example, to enable the Performance Schema but exclude stage and statement instrumentation, do this:

  ```
  shell> cmake . -DWITH_PERFSCHEMA_STORAGE_ENGINE=1 \
          -DDISABLE_PSI_STAGE=1 \
          -DDISABLE_PSI_STATEMENT=1
  ```

  For more information, see the descriptions of the `DISABLE_PSI_XXX` `CMake` options in MySQL Source-Configuration Options. (Bug #17478068)

- A new `CMake` option, `WITH_ASAN`, permits enabling AddressSanitizer for compilers that support it. (Bug #17435338)

- Several compilation warnings were fixed that occurred when compiling without debugging enabled. (Bug #17332094)

- The implementation of condition variables specific to Windows XP and Windows Server 2003 was removed from the source code because MySQL is not supported on those platforms as of MySQL 5.6. (Bug #17332056)

- A new `ER_ENGINE_OUT_OF_MEMORY` error code is available for use by storage engines to report out-of-memory conditions. (Bug #16807964)

- For `GRANT` statements, `ER_SP_DOES_NOT_EXIST` errors for nonexistent stored procedures and functions now specify `PROCEDURE does not exist` or `FUNCTION does not exist` rather than the less-specific `PROCEDURE or FUNCTION does not exist`. (Bug #69628, Bug #17036976)

- The hash function used for metadata locking was modified to reduce overhead. (Bug #68487, Bug #16396598)

- Overhead for deprecation warnings was reduced. (Bug #70402, Bug #17497869)

- The optimizer now is able to apply the range scan access method to queries of this form:

  ```
  SELECT ... FROM t1 WHERE ( col_1, col_2 ) IN (( 'a', 'b' ), ( 'c', 'd' ));
  ```

  Previously, for range scans to be used it was necessary for the query to be written as:

  ```
  SELECT ... FROM t1 WHERE ( col_1 = 'a' AND col_2 = 'b' )
  OR ( col_1 = 'c' AND col_2 = 'd' );
  ```

  For the optimizer to use a range scan, queries must satisfy these conditions:

  - Only `IN` predicates can be used, not `NOT IN`.

  - There may only be column references in the row constructor on the `IN` predicate's left hand side.

  - There must be more than one row constructor on the `IN` predicate's right hand side.

  - Row constructors on the `IN` predicate's right hand side must contain only runtime constants, which are either literals or local column references that are bound to constants during execution.

  `EXPLAIN` output for applicable queries will change from full table or index scan to range scan. Changes are also visible by checking the values of the `Handler_read_first`, `Handler_read_key`, and `Handler_read_next` status variables.

- When a connection is returned to the thread pool plugin, the connection thread context must be cleaned up. Previously, this was done using `COM_CHANGE_USER` (which is like the `mysql_change_user()` C API function). However, that operation reauthenticates, which is unnecessary network roundtrip overhead in this context.

  Now it is possible for client connection state to be reset in a more lightweight manner without causing reauthentication. The API is exposed publicly through these changes:

  - A new `COM_RESET_CONNECTION` protocol command (defined in `mysql_com.h`)

  - A new `mysql_reset_connection()` C API function

  - A new `resetconnection` command for the `mysql` client

Resetting a connection has effects similar to mysql_change_user() or an auto-reconnect except that the connection is not closed and reopened, and reauthentication is not done. See mysql_change_user()) and see Controlling Automatic Reconnection Behavior).

For more information, see mysql_reset_connection() and mysql — The MySQL Command-Line Tool.

- A new mysql_get_option() C API function is available that returns the current value of applicable mysql_options() options. See mysql_get_option().
- Because there are new API functions (mysql_get_option(), mysql_reset_connection()), the library ABI version is now 18.2. Shared library names now include 18.2 where appropriate.

Bugs Fixed

- **Incompatible Change:** For logging to the general_log and slow_log tables in the mysql database, log lines containing multiple character sets were not always handled correctly. The general_log.argument and slow_log.sql_text columns now have been changed from MEDIUMTEXT to MEDIUMBLOB. Consequently, no character set transformation is done for logging to tables now, which aligns it with logging to files.

  If you upgrade to this release of MySQL from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate these changes into the mysql database. (Bug #14575847)
- **Important Change; Replication:** START SLAVE UNTIL SQL_AFTER_GTIDS did not cause the slave to stop until the next GTID event was received following execution of the transaction having the indicated GTID, which could cause issues in the case when the next GTID event is delayed, or does not exist. Now the slave stops after completing the transaction with that GTID. (Bug #14767986)
- **Important Change; Replication:** It was possible to start the server with binary logging enabled but no server ID specified; in such cases, the server would set server_id to 1 (rather than 0) while slaves remained unable to connect.

  Now --server-id must be used when starting the server with binary logging enabled, otherwise the server is unable to start. If --server-id=0 is used, this value is no longer changed by the server; in this case, updates are written to the binary log, but slaves are unable to connect. Using --server-id without specifying a value has the same effect as setting it explicitly to 0. (Bug #11763963, Bug #56739)
- **InnoDB:** When new indexes are added by an ALTER TABLE operation, instead of only saving table-level statistics and statistics for the new indexes, InnoDB would save statistics for the entire table, including the table's other indexes. This behavior slowed ALTER TABLE performance. (Bug #17848838, Bug #16511145)
- **InnoDB:** A regression introduced by the fix for Bug#17371537 resulted a memory leak for memcached insert operations. (Bug #17738935)
- **InnoDB:** The trx->error_key_num field was not initialized in the error injection code found in storage/innobase/handler/handler0alter.cc. The error_key_num field is usually 0 but can be a non zero value if the memory buffer of a DDL transaction object is reused. (Bug #17624926)
- **InnoDB:** Fault-tolerant code found in the log apply code for InnoDB ALTER TABLE ... IN PLACE could result in data corruption. (Bug #17625063, Bug #17512497)
- **InnoDB:** Databases names beginning with a digit would cause a full-text search (FTS) parser error. (Bug #17607956)

  References: See also Bug #17161372.

- **InnoDB:** An ALTER TABLE ... CHANGE [COLUMN] operation would result in an rbt_empty(index_cache->words) assertion. (Bug #17536995)
- **InnoDB:** Running SHOW ENGINE INNODB STATUS on one connection thread and killing that thread by running a KILL CONNECTION statement from a different connection thread would result in a severe error. (Bug #17474166)
- **InnoDB:** An excessive amount of memory would be consumed when querying INFORMATION_SCHEMA.INNODB_FT_INDEX_TABLE. The problem would occur for very large full-text search indexes. (Bug #17483582, Bug #70329)
- **InnoDB:** buf_flush_event would be created at flush thread startup instead of server startup. Also, buf_flush_event would be signaled when InnoDB is started in read-only mode. (Bug #17516062)
- **InnoDB:** CHECK TABLE would ignore the QUICK option. (Bug #17513737)
- **InnoDB:** In debug builds, test case failures would occur due to ibuf_contract_ext performing merges and dict_stats_update returning evicted pages back into the buffer pool while ibuf_change_buffering_debug is enabled. (Bug #17446090)
- **InnoDB:** InnoDB would fail to return an error when attempting to run a query after discarding the tablespace. (Bug #17431533)
- **InnoDB:** Data in the OPERATION column of performance_schema.events_waits_current table was incorrect due to a code regression introduced in MySQL 5.7.2. (Bug #17429480)
- **InnoDB:** A severe error would occur after discarding a tablespace. (Bug #17430207)
- **InnoDB:** During a TRUNCATE TABLE operation, InnoDB: Trying to TRUNCATE a missing index of table ... warnings would be printed to the error log. These warnings should not be printed when the index is a full-text search (FTS) index. (Bug #17402002, Bug #70226)

  References: See also Bug #12429565.
- **InnoDB:** The UNIV_BLOB_DEBUG compile-time debug flag, which is not often used and is limited by its inability to work across crash recovery, has been removed in order to simplify code refactoring. (Bug #17338452)
- **InnoDB:** The UNIV_SEARCH_DEBUG compile-time debug check, which has never reported a failure, has been removed along with all references to it. (Bug #17338432)
- **InnoDB:** An assertion would be raised when the database initialization thread encountered other threads performing buffer pool flushing. (Bug #17349975)
- **InnoDB:** Full-text index creation on a large table would fail due to insufficient temporary table space and result in a misleading “incorrect key file” error. (Bug #17339606)
- **InnoDB:** row_scan_index_for_mysql would allocate a buffer size of UNIV_PAGE_SIZE for row_search_for_mysql. When the record length was greater than UNIV_PAGE_SIZE, a Valgrind error would occur. (Bug #17378106)
- **InnoDB:** The InnoDB mecached plugin would update a record before inserting to the binlog, which would cause slave server replication to stop. The insert should occur before the update. (Bug #17358875)
- **InnoDB:** During parallel full-text search (FTS) index creation, a scanner thread reads in documents and passes them to the tokenizer. The tokenizer frees documents from memory when tokenization is complete. When tokenizing documents with a large amount of text, the tokenizer thread would not keep

pace with the scanner thread. As a result, memory would not be freed fast enough and the “tokenization pending list” would grow in size. (Bug #17384979)

- **InnoDB:** A full-text search (FTS) BOOLEAN MODE query with an invalid character in the query string could result in a memory access violation failure. (Bug #17350055)

- **InnoDB:** trx_create and trx_free would be called for every memcached get request. (Bug #17371537, Bug #70172)

- **InnoDB:** When innodb_file_per_table is set to OFF, replication would fail with error code 1880 when truncating tables. For tables that reside in the shared tablespace, the truncate log file name used during the truncate action was not unique. The error is due to a MySQL 5.7.2 code regression related to Bug #14174004. (Bug #17327409)

- **InnoDB:** trx_sys_t::ro_trx_list has been removed. Adding and removing transactions from the ro_trx_list can be costly, and placing transactions on the list by default is no longer necessary after mutex related optimizations implemented in MySQL 5.7.2. User transactions are still placed on the trx_t::mysql_trx_list. Currently, background read-only transactions are not placed on any list. (Bug #17332300)

- **InnoDB:** When InnoDB is retrieving rows and a KILL QUERY statement is issued, InnoDB would return false errors. (Bug #16950658)

- **InnoDB:** The hardcoded size for the srv_max_n_threads variable was insufficient. The variable setting is now configured based on the maximum number of connection threads and InnoDB background threads. (Bug #16884077)

- **InnoDB:** In btr_validate_level there are checks to ensure that all B-tree pages are marked when allocated. The checks would fail on the change buffer because the allocation of change buffer pages is handled differently than other B-tree pages. (Bug #16884217)

- **InnoDB:** InnoDB would set UNIV_WORD_SIZE to 4 for both Windows 32-bit and 64-bit systems. With this patch, UNIV_WORD_SIZE is set to 8 on 64-bit Windows systems. This patch also removes UNIV_WORD_ALIGNMENT, which is no longer used. (Bug #16774645)

- **InnoDB:** When the change buffer is enabled, InnoDB would fail to write a transaction log record when merging a record from the insert buffer to a secondary index page if the insert was performed as an “update-in-place”. (Bug #16752251, Bug #69122)

- **InnoDB:** A SELECT COUNT(*) query would take a long time to complete when run concurrently with a LOAD DATA operation. The mtr_memo_contains function, which determines if an object is part of a memo in a mini transaction, contained a nested loop that caused the query to run slowly. (Bug #16764240, Bug #69141)

- **InnoDB:** Due to a regression in MySQL 5.6, creating or dropping tables with innodb_force_recovery set to 3 (SRV_FORCE_NO_TRX_UNDO) would fail. Additionally, this bug fix includes a code modification that sets InnoDB to read-only when innodb_force_recovery is set to a value greater than 3 (SRV_FORCE_NO_TRX_UNDO). (Bug #16631778, Bug #69892)

- **InnoDB:** An InnoDB memcached configuration error message contained an incorrect file name. The error message stated, Please create config table containers in database innodb_memcache by running innodb_config.sql. error 31. The correct file name is innodb_memcached_config.sql. Also, the “error 31” portion of the error message has been translated to its text equivalent, which is “Table not found”. (Bug #16498810, Bug #68684)

- **InnoDB:** In mutex_spin_wait(), the sync_array_reserve_cell function could fail to find an empty slot on systems with sync wait arrays that are small in size, resulting in an error. (Bug #16245498)

- **InnoDB:** When `index_read_map` is called for an exact search and fails to return a record due to non-matching search criteria, the cursor would be positioned on the next record after the searched key. A subsequent call to `index_next` would return the next record instead of returning the previous non-matching row, thereby skipping a record. (Bug #14621190, Bug #15965874, Bug #17314241, Bug #70038, Bug #17413093, Bug #12860669, Bug #60220, Bug #17565888)

- **InnoDB:** An infinite loop could occur in `buf_page_get_gen` when handling compressed-only pages. (Bug #12560151, Bug #61132)

- **InnoDB:** `InnoDB` would attempt to free `BLOB` pages already marked as free. (Bug #11762662, Bug #55284)

- **InnoDB:** Converting a table with a large number of columns from `MyISAM` to `InnoDB` would cause an assertion due to insufficient log buffer space. Instead of asserting, `InnoDB` now attempts to increase log buffer size automatically if the redo log size is too large. (Bug #11758196, Bug #50366)

- **Partitioning:** The storage engine was set incorrectly during a rebuild of a partition; the table storage engine was ignored and the default storage engine used instead. Thus, in MySQL 5.1, it was possible for `REBUILD PARTITION` to change the partition storage engine from `InnoDB` to `MyISAM`, and for the reverse (rebuilding partitions of `MyISAM` tables causing the partitions to use `InnoDB`) to occurin MySQL 5.5 and later. Now, when rebuilding partitions, the storage engine actually used by the table is checked and used by the handler for the rebuild operation, so that the partition storage engine is not inadvertently changed. (Bug #17559867)

- **Partitioning:** Index condition pushdown did not work with partitioned tables. (Bug #17306882, Bug #70001)

- **Partitioning:** After disabling the parent table's indexes with `ALTER TABLE ... DISABLE KEYS`, rebuilding any of its partitions enabled the indexes on those partitions, leading `MyISAM` to fail with an error when the optimizer tried to use one of the affected indexes.

  Now in such cases, we check for disabled indexes on the table before rebuilding any of its partitions. If the indexes have been disabled, then we disable them on the partition following the rebuild. (Bug #16051817)

- **Replication:** When GTID-based replication was used with an intra-schema multi-threaded slave, transactions were assigned to the first worker thread only. (Bug #17590616, Bug #70536)

- **Replication:** The `WORKER_ID` column of the `replication_execute_status_by_worker` table did not use the internal `id` column values from the `mysql.slave_worker_info` table, as expected. (Bug #17514406, Bug #70426)

- **Replication:** The `THREAD_ID` column values shown in the `performance_schema.replication_execute_status_by_worker` table used the same thread IDs shown in the output from `SHOW PROCESSLIST`, rather than those used by other `performance_schema` tables. (Bug #17440991, Bug #70423)

  References: See also Bug #17473308, Bug #17526982.

- **Replication:** The final argument in the `SET` clause of a `LOAD DATA ... SET` statement was repeated in the binary log. (Bug #17429677, Bug #70277)

- **Replication:** When an error encountered by the dump thread while reading events from the active binary log file was a temporary error, so that the dump thread tried to read the event, it was possible for the dump thread to seek the wrong position, which could cause one or more events to be resent. To prevent this, the thread's position is obtained after each correct read of an event.

In addition, with this fix, only binary logs that are not closed normally are marked as possibly being corrupted.

Finally, two warnings are added; these are now returned when a dump thread encounters a temporary error. (Bug #17402313)

- **Replication:** When stopping the I/O thread, it was possible with a very large transaction (equivalent to a binary log size greater than 100MB) that the thread did not receive the transaction to the end. When reconnecting with MASTER_AUTO_POSITION=1 it then tried to fetch changes from the next transaction, which could lead to loss of the incomplete transaction and its data. (Bug #17280176, Bug #69943)

- **Replication:** Setting rpl_semi_sync_master_enabled while the master was waiting for a reply from the slave could in some cases cause the master to fail. (Bug #17327454, Bug #70045)

- **Replication:** Trying to set CHANGE MASTER TO ... MASTER_AUTO_POSITION = 0 failed with error 1777 (**ER_AUTO_POSITION_REQUIRES_GTID_MODE_ON**). (Bug #17277744)

- **Replication:** No error was written to the log file when writing an incident event to the binary log. This meant that the user was not alerted that a slave server will later be stopped by the incident event. To prevent this from happening, error messages are now written in the log file for all incidents written to the binary log. (Bug #17258782)

- **Replication:** The value of LAST_INSERT_ID() was not correctly replicated when filtering rules were used on the slave. (Bug #17234370, Bug #69861)

- **Replication:** RESET SLAVE ALL reset only the host, port, user, password, and log positions. Now it resets all connection parameters. (Bug #17185647)

- **Replication:** An internal function used for storing GTID values could sometimes try to handle them as strings of the wrong length. (Bug #17032712, Bug #69618)

- **Replication:** During row-based replication with binlog_row_image set to MINIMAL, updating only some columns of a table having 9 or more columns caused mysqlbinlog to fail when it was used with the --verbose option. (Bug #16960133)

- **Replication:** When a master with semisynchronous replication enabled was shut down, the master failed to wait for either a semisyncnronous ACK or timeout before completing the shutdown. This prevented semisynchronous replication from reverting to asynchronous replication and allowed open transactions to complete on the master, which resulted in missing events on the slave.

  To fix this problem, dump threads are now stopped last during shutdown, after the client is told to stop, so that, if the dump thread has pending events from active clients, they can be sent to the slave. (Bug #16775543)

- **Replication:** Issuing a GRANT statement with invalid parameters caused the master to write LOST_EVENTS events into its binary logs, causing replication to stop. Now such cases, if one or more grants or revocations of privileges are successful, an incident is written to the log; otherwise, only a warning is logged. (Bug #16629195, Bug #68892)

- **Replication:** START SLAVE failed when the server was started with the options --master-info-repository=TABLE relay-log-info-repository=TABLE and with autocommit set to 0, together with --skip-slave-start.

  A workaround for previous versions of MySQL is to restart the slave mysqld without the --skip-slave-start option. (Bug #16533802)

- **Replication:** A number of unneeded initializations of objects that were used but not actually needed for reading the relay info log were removed. (Bug #16291602)
- **Replication:** mysqlbinlog now supports the same command-line options relating to SSL as mysql, mysqladmin, and other MySQL client programs. See SSL Command Options, for more information. (Bug #11751199, Bug #41975)
- UNION ALL statements for which one SELECT returned zero rows could result in an incorrect FOUND_ROWS() value. (Bug #17708480)
- For debug builds, an error occurring during DELETE IGNORE could raise an assertion. (Bug #17720294)
- Enabling Index Merge optimizer switches and setting a small sort_buffer_size value could lead to a server exit. (Bug #17617945)
- CMake configuration for the Clang compiler sets more appropriate flags for building on Linux. Specifically, -g -fno-omit-frame-pointer -fno-strict-aliasing is now added. (Bug #17633291)
- Disconnect processing overhead was reduced for sessions that have no outstanding prepared statements. Previously, a global mutex was acquired to handle these, but there is no need to do so in the absence of such statements. (Bug #17560986)
- Semi-join materialization strategy was not used for VARCHAR columns longer than 512 bytes, resulting in use of a less-efficient strategy and worse query performance. (The limit in characters rather than bytes depends on the column character set; 170 characters for utf8, for example.) (Bug #17566396)
- UNION ALL queries for which the last SELECT selected an aggregate value from an empty table resulted in an incorrect FOUND_ROWS() value. (Bug #17580869)
- Some license and documentation files were missing from Windows MSI packages. (Bug #17584523)
- The optimizer did not consider a clustered primary key as a covering index unless all columns required for a query were in the primary key definition. This incorrectly raised the calculated cost of using the index and caused it not to be used in some cases when it would be more efficient than the index otherwise chosen. (Bug #17560636)
- Selecting from the Performance Schema session_connect_attrs table under high load could cause a server exit. (Bug #17542370)
- Performance Schema memory instrumentation overhead was reduced.

  Memory allocated internally by the Performance Schema was not reported by the memory instrumentation. This was corrected, and there is now a memory/performance_schema/internal_buffers instrument (disabled by default) that can be enabled to obtain this information. (Bug #17493868)
- Stored programs were not listed in the Performance Schema objects_summary_global_by_type table. (Bug #17472833)
- The Performance Schema had a race condition adding and deleting stored procedure entries that could raise an assertion. (Bug #17529279)
- my_print_stacktrace() printed a Reference Manual URL that pointed to the 5.1 manual. It now is parameterized for the current server series. (Bug #17465503)
- DROP TRIGGER succeeded even with the read_only system variable enabled. (Bug #17503460)

- An incorrect result could be produced by a simple COUNT(DISTINCT) query on a table that contains a large number of distinct values. This was more likely when tmp_table_size or max_heap_table_size were set to small values. (Bug #17500866)
- An assertion was raised if a statement tried to set an exception condition in a diagnostics area which already contained an exception or completion condition. This could occur if a failed stored program tried to transfer its exception condition to the diagnostics area of its caller that already contained a condition. (Bug #17400687)
- In the Performance Schema memory instrumentation, statistics collected for memory-free operations could be under-evaluated, leading to the appearance of a memory leak. (Bug #17400029)
- KILL with an invalid thread ID value could raise an assertion. (Bug #17420682)
- Compilation failures under Visual Studio 2012 were corrected. (Bug #17430236)
- For JSON-format EXPLAIN statements, materialized views were incorrectly labeled as optimized_away_subqueries rather than as materialized_from_subquery. (Bug #17428655)
- In debug builds, SHOW PROCEDURE CODE raised an assertion. (Bug #17434385)
- The mysql_real_connect() C API function could leak memory if it failed. (Bug #17337684)
- Savepoints could not be used successfully following an ER_LOCK_DEADLOCK error (or ER_LOCK_WAIT_TIMEOUT error, if innodb_rollback_on_timeout was enabled). (Bug #17356954)

  References: This bug is a regression of Bug #14188793.
- The CLIENT_CONNECT_WITH_DB flag was improperly handled in the C client library. This could lead to a malformed packet sent to the server. (Bug #17351732)
- Upgrades using RPM packages failed if the MySQL server was running due to failure to ignore the mysqld_safe.pid file created by mysqld_safe. (Bug #17343851)
- Compiling without the Performance Schema resulted in unresolved symbols. (Bug #17399658)
- An assertion was raised if SET PASSWORD was used for an account that has been manually deleted from the mysql.user table but still present in memory. (Bug #17359329)
- Some warnings produced by mysql_install_db referred to the now-deprecated log_warnings system variable. These have been updated to refer to log_error_verbosity instead. (Bug #17363350)
- Full-text search on InnoDB tables failed on searches that used the + boolean operator. (Bug #17280122)
- The mysql_options() C API function could leak memory if called more than once with the MYSQL_SET_CLIENT_IP option. (Bug #17297012)
- The CONV() function could call abs(INT_MIN), which is undefined, and cause a server exit. (Bug #17296644)
- An error array in the SSL code was missing a comma, leading to implicit concatenation of adjacent messages and a resulting off-by-one error in the relationship between error numbers and messages. (Bug #17294150)
- Cost comparison for ref access could be inaccurate. (Bug #17303649)
- The filesort implementation sometimes failed to allocate enough buffer space, leading to a server exit. (Bug #17326567)

- For single-threaded workloads, the optimizer recognizes some special cases for which it can avoid function calls and enhance performance. (Bug #17234723)
- The my_load_path() function could in some cases calculate a path value incorrectly. (Bug #17204851)
- Within a stored program, comparison of the value of a scalar subquery with an IN clause resulted in an error for the first execution and raised an assertion for the second execution. (Bug #17029399)
- In debug builds, an assert could be raised if a statement was killed while executing a trigger. (Bug #17049537)
- Temporary table columns were marked as temporarily nullable without taking into account the presence of triggers for the table. This could lead to NOT NULL columns being updated to NULL by a multiple-table UPDATE statement. (Bug #17055378)
- JSON-format EXPLAIN statements could leak memory. (Bug #16970785)
- Some queries with ROLLUP and an inner subquery with a reference to an outer field could raise an assertion. (Bug #16967281)
- Information was not transferred between two optimizer modules because there were duplicate variables for the same information. This could lead to suboptimal query execution plans and incorrect query results. (Bug #16982071, Bug #70021, Bug #17310161)
- GRANT without an IDENTIFIED BY clause resulted in an error even for existing users. (Bug #16938568)
- For partitioned tables, queries could return different results depending on whether Index Merge was used. (Bug #16862316)

  References: See also Bug #17648468, Bug #176588348, Bug #18167648.
- GROUP_CONCAT() with an invalid separator could cause a server exit. (Bug #16870783)
- mysqltest_embedded does not work with the --ps-protocol option, so it now issues a warning if that option is given. (Bug #16817580)
- An internal InnoDB string routine could write past the end of a buffer. (Bug #16765410)
- GIS intersection-related code was missing a return value check, leading to a loop in nondebug builds and a raised assertion in debug builds. (Bug #16659166)
- For debug builds, when the optimizer removed an Item_ref pointing to a subquery, it caused a server exit. (Bug #16509874)

  References: This bug is a regression of Bug #16318585.
- It is no longer possible to expire the password for anonymous-user accounts because an anonymous user cannot execute SET PASSWORD to reset the account password. (Bug #16483619)
- On Windows, a MySQL client program that simply used #include <mysql.h> could not be compiled due to missing Windows declarations in that file. The same program would compile on other platforms. (Bug #16409270)

  References: See also Bug #17514554.
- HANDLER READ could cause a server exit due to wrongly considering columns as constant. (Bug #16386136)

- Using the binary client/server protocol, the second execution of a prepared statement for a query with parameters in the `LIMIT` clause raised an assertion. (Bug #16346241)

- Very long database names in queries could cause the server to exit. (Bug #15912213, Bug #16900358)

- Memory allocated for the Performance Schema was not freed at server shutdown. (Bug #14771682)

- If an `INSTALL PLUGIN` statement contained invalid UTF-8 characters in the shared library name, it caused the server to hang (or to raise an assertion in debug builds). (Bug #14653594)

- Standalone Windows MSI packages did not have the `ALLUSERS` property set. They now set `ALLUSERS=1`. For earlier MSI packages in this MySQL series, a workaround is to use the following command:

  ```
  C:\> msiexec /i msi_installer_name ALLUSERS=1
  ```

  (Bug #14647206)

- Deadlocks involving metadata locks and `InnoDB` deadlocks were both reported as an `ER_LOCK_DEADLOCK` error, but only `InnoDB` deadlocks rolled back the transaction. Now both deadlocks roll back the transaction. (Bug #14188793)

- Columns in a `PRIMARY KEY` must be `NOT NULL`, but if declared explicitly as `NULL` produced no error. Now an error occurs. For example, a statement such as `CREATE TABLE t (i INT NULL PRIMARY KEY)` is rejected. The same occurs for similar `ALTER TABLE` statements. (Bug #13995622, Bug #66987, Bug #15967545, Bug #16545198)

- Fixed a potential problem with the MySQL string function `strmov()`. Its behavior with respect to overlap of source and destination previously depended on native support for `stpcpy()`. (Bug #48864, Bug #17429539)

- If one connection changed its default database and simultaneously another connection executed `SHOW PROCESSLIST`, the second connection could access invalid memory when attempting to display the first connection's default database. memory. (Bug #58198, Bug #11765252)

- Full-text search on `InnoDB` tables failed on searches for words containing apostrophes when using boolean operators. (Bug #69932, Bug #17276125)

- `InnoDB` deadlock caused transaction rollback but did not release metadata locks, blocking concurrent DDL on the transaction tables until the connection that got the deadlock issued an explicit `COMMIT` or `ROLLBACK`. (Bug #69668, Bug #17054007)

- In debug builds, static initialization code could call DBUG functions before the DBUG subsystem was initialized. (Bug #69653, Bug #17063675)

- With the thread pool plugin enabled, the `PROCESSLIST_USER` and `PROCESSLIST_HOST` columns of the Performance Schema `threads` table were always `NULL` for client sessions. Also, for the main thread, those columns were not `NULL` but set to a user account.

  > **Note**
  >
  > As part of the bug fix implementation, Performance Schema instrumentation for the thread pool plugin was changed to use `thread_pool`, not `sql`.

  (Bug #70028, Bug #17310065, Bug #17049691)

- `COUNT(DISTINCT)` should not count `NULL` values, but they were counted when the optimizer used Loose Index Scan. (Bug #69841, Bug #17222452)

- For queries of the form `UPDATE ... WHERE` *`unique_key`* `ORDER BY ... LIMIT ...`, incorrect rows could be updated. Unique keys permit multiple `NULL` values, but the optimizer did not always consider all of them. (Bug #68656, Bug #16482467)
- If asked to upgrade a server that was running without `InnoDB` enabled, `mysql_upgrade` issued complaints about `InnoDB` tables not existing (tables that will not exist unless `InnoDB` is available). (Bug #70152, Bug #17361912)
- In some cases, range conditions over indexes defined on column prefixes returned incomplete result sets. (For example, `SELECT ... WHERE 'abcdef1' <` *`col_name`* `AND` *`col_name`* `< 'abcdef9'`, where the index on *`col_name`* indexed only the first 6 characters.) (Bug #70341, Bug #17458273)
- Reads of `Geometry` values within a stored program could read already-freed memory and produce incorrect results. (Bug #69517, Bug #16985214)
- `ER_TRUNCATED_WRONG_VALUE` (truncated value) errors for `DECIMAL` values failed to show the erroneous input value. (Bug #68745, Bug #16552002)
- Performance Schema instrumentation overhead was reduced for frequent connect/disconnect operations. (Bug #70018, Bug #17310878)
- For a column declared as a `PRIMARY KEY`, the MySQL sometimes unnecessarily added a `DEFAULT` clause. For example, for `CREATE TABLE t (a INT, PRIMARY KEY(a))`, a `DEFAULT 0` clause was added, whereas with `CREATE TABLE t (a INT PRIMARY KEY)`, it was not. (Bug #36723, Bug #11748566)
- Setting `host_cache_size` at startup had no effect. (Bug #70552, Bug #17576516)
- `InnoDB` full-text searches failed to find records within transactions that included savepoints. (Bug #70333, Bug #17458835)
- Some `INSERT INTO ... SELECT ... FROM` statements were slow unless the `tmp_table_size` and `max_heap_table_size` system variables were set large enough to permit the temporary table used for query processing to be stored in the `MEMORY` storage engine. (Bug #69368, Bug #16894092)
- Incorrect reference counting in the range optimizer module resulted in potential for missing or duplicate rows in the query result set. (Bug #70236, Bug #17405466)
- With a `NULL` literal in a particular parameter position, `IFNULL()`, `COALESCE()`, and `CASE` returned a signed value when they should return an unsigned value. (Bug #65976, Bug #14359340)
- For queries that qualify to be written to the slow query log, the check whether the log is enabled now occurs earlier, reducing overhead.

  Also, when `log_throttle_queries_not_using_indexes` is specified at server startup, a value is now required. Previously, it was incorrectly treated as optional. (Bug #69420, Bug #16924125)
- Host names in grant tables are stored in lowercase, but `mysql_install_db` could fail to observe this convention, leading to accounts that could not be dropped with `DROP USER`. (Bug #62255, Bug #12917164, Bug #62254, Bug #12917151)
- For `EXPLAIN FOR CONNECTION` *`connection_id`*, the parser did not permit *`connection_id`* values larger than the maximum signed `BIGINT` value. The maximum unsigned value now is permitted. (Bug #70533, Bug #17564492)
- At server shutdown, a race condition between the main thread and the shutdown thread could cause server failure. (Bug #56666, Bug #11763896)

- `EXPLAIN FOR CONNECTION` reported no error if the connection ID belonged to no connection thread. (Bug #70532, Bug #17564493)

- The server uses the ethernet hardware address for UUID generation, but made assumptions about the names of ethernet devices rather than querying the system for their names. Thanks to Honza Horak for the patch. (Bug #63055, Bug #13548252)

- Several issues identified by the Coverity static analysis tool were fixed. Thanks to Jan Staněk and Honza Horak for the patches. (Bug #70591, Bug #17590095)

- Missing `va_end()` calls were added to logging and UCS2 code. Thanks to Jan Staněk for the patch. (Bug #68896, Bug #16725769)

- MySQL did not compile on Mac OS X 10.9 (Mavericks). (Bug #70542, Bug #17647863)

- Within a stored function or trigger, occurrence of an `ER_LOCK_WAIT_TIMEOUT` error caused failure to execute a condition handler. (Bug #67947, Bug #16041903)

- Specifying the same directory with multiple instances of `--ignore-db-dir` caused a server exit. (Bug #69441, Bug #16944177)

- Killing a query that is performing a `filesort` operation resulted in an `ER_SERVER_SHUTDOWN` (Server shutdown in progess) error. (Bug #18256, Bug #11745656)

- Views containing `ORDER BY` *`integer`* could result in errors at view evaluation time. Consider these view definitions, which use `ORDER BY` with an ordinal number:

  ```
  CREATE VIEW v1 AS SELECT x, y, z FROM t ORDER BY 2;
  CREATE VIEW v2 AS SELECT x, 1, z FROM t ORDER BY 2;
  ```

  In the first case, `ORDER BY 2` refers to a named column `y`. In the second case, it refers to a constant 1. For queries that select from either view fewer than 2 columns (the number named in the `ORDER BY` clause), an error occurred if the server evaluated the view using the MERGE algorithm. Examples:

  ```
  mysql> SELECT x FROM v1;
  ERROR 1054 (42S22): Unknown column '2' in 'order clause'
  mysql> SELECT x FROM v2;
  ERROR 1054 (42S22): Unknown column '2' in 'order clause'
  ```

  To handle view definitions like this, the server now writes them differently into the `.frm` file that stores the view definition. This difference is visible with `SHOW CREATE VIEW`. Previously, the `.frm` file contained this for the `ORDER BY 2` clause:

  ```
  For v1: ORDER BY 2
  For v2: ORDER BY 2
  ```

  Now, the `.frm` file contains this:

  ```
  For v1: ORDER BY `t`.`y`
  For v2: ORDER BY ''
  ```

  That is, for `v1`, 2 is replaced by a reference to the name of the column referred to. For `v2`, 2 is replaced by a constant string expression (ordering by a constant has no effect, so ordering by any constant will do).

If you experience view-evaluation errors such as just described, drop and recreate the view so that the `.frm` file contains the updated view representation. Alternatively, for views like `v2` that order by a constant value, drop and recreate the view with no `ORDER BY` clause. (Bug #28695, Bug #11746789)

Changes in MySQL 5.7.2 (2013-09-21, Milestone 12)

> **Note**
>
> This is a milestone release, for use at your own risk. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward.

Authentication Notes

- **Incompatible Change:** Previously, account rows in the `mysql.user` table could have an empty `plugin` column value. In this case, the server authenticated such an account using either the `mysql_native_password` or `mysql_old_password` plugin, depending on whether the password hash value in the `Password` column used native hashing or the older pre-4.1 hashing method. With the deprecation of old-format password hashes in MySQL 5.6.5, this heuristic for deciding which authentication plugin to use is unnecessary and it is desirable that `user` table rows always specify explicitly which authentication plugin applies.

  To that end, the `plugin` column is now defined to be non-`NULL` with a default value of `'mysql_native_password'`, and associated server operations require the column to be nonempty. In conjunction with this `plugin` column definition modification, several other changes have been made:

  - The `--default-authentication-plugin` command-line option is reimplemented as the `default_authentication_plugin` system variable. Its use at server startup is unchanged, but now the default plugin value can be examined at runtime using `SHOW VARIABLES` or `SELECT @@default_authentication_plugin`. The variable is read only and cannot be changed at runtime.

  - When `mysql_install_db` is run, it invokes the server to initialize the `mysql` database. The server now assigns every `user` table row a nonempty `plugin` column value. The value is `'mysql_native_password'` unless the `default_authentication_plugin` system variable is set otherwise at server startup.

  - `mysql_upgrade` checks `user` table rows and, for any row with an empty `plugin` column, sets that column to `'mysql_native_password'` or `'mysql_old_password'` depending on the hash format of the `Password` column value.

  - At startup, and at runtime when `FLUSH PRIVILEGES` is executed, the server checks `user` table rows. For any row with an empty `plugin` column, the server writes a warning to the error log of this form:

    ```
    [Warning] User entry 'user_name'@'host_name' has an empty plugin
    value. The user will be ignored and no one can login with this user
    anymore.
    ```

    To address this issue, execute `mysql_upgrade`.

  If you upgrade to this release of MySQL from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate the `plugin` column change into the `mysql` database and assign the appropriate nonempty plugin value to any empty `plugin` column values. However, because the server now checks for and disables accounts with empty `plugin` column values, it is necessary to upgrade as follows.

For an upgrade in which you plan to use the data directory from your existing MySQL installation:

1. Stop the server
2. Upgrade MySQL in place
3. Restart the server with the `--skip-grant-tables` option to disable privilege checking
4. Run `mysql_upgrade`
5. Restart the server normally (without `--skip-grant-tables`)

For an upgrade in which you plan to reload a dump file generated from your existing MySQL installation:

1. To generate the dump file, run `mysqldump` without the `--flush-privileges` option
2. Stop the server
3. Upgrade MySQL in place
4. Restart the server with the `--skip-grant-tables` option to disable privilege checking
5. Reload the dump file (`mysql < *dump_file*`)
6. Execute `mysql_upgrade`
7. Restart the server normally (without `--skip-grant-tables`)

`mysql_upgrade` runs by default as the MySQL `root` user. For either of the preceding procedures, if the `root` password is expired when you run `mysql_upgrade`, you will see a message that your password is expired and that `mysql_upgrade` failed as a result. To correct this, reset the `root` password to unexpire it and run `mysql_upgrade` again:

```
shell> mysql -u root -p
Enter password: ****  <- enter root password here
mysql> SET PASSWORD = PASSWORD('root-password');
mysql> quit

shell> mysql_upgrade
```

`SET PASSWORD` normally does not work if the server is started with `--skip-grant-tables`, but the first invocation of `mysql_upgrade` flushes the privileges, so when you run `mysql`, the `SET PASSWORD` statement is accepted.

Diagnostics Notes

- **Incompatible Change:** Per the SQL standard, nondiagnostic statements should clear the diagnostics area when they begin executing. Previously, MySQL differed from this in that some nondiagnostic statements did not do this. MySQL now follows the SQL standard, which affects the content of the diagnostics area for some statements. Consequently, the result from statements such as `SHOW WARNINGS` that display the diagnostics area now differs somewhat:
  - The previous behavior: `SHOW WARNINGS` displays information about the conditions (errors, warnings, and notes) resulting from the most recent statement in the current session that generated messages. It shows nothing if the most recent statement used a table and generated no messages. (That is,

statements that use a table but generate no messages clear the message list.) Statements that do not use tables and do not generate messages have no effect on the message list.

- The new behavior: SHOW WARNINGS displays information about the conditions resulting from execution of the most recent nondiagnostic statement in the current session.

The result from other diagnostic statements is affected similarly (SHOW ERRORS, GET DIAGNOSTICS).

The following example demonstrates the difference in behavior.

Previously:

```
mysql> DROP TABLE test.no_such_table;
ERROR 1051 (42S02): Unknown table 'test.no_such_table'
mysql> SELECT @@warning_count;
Query OK, 0 rows affected (0.00 sec)
mysql> SHOW WARNINGS;
+-------+------+------------------------------------+
| Level | Code | Message                            |
+-------+------+------------------------------------+
| Error | 1051 | Unknown table 'test.no_such_table' |
+-------+------+------------------------------------+
1 row in set (0.00 sec)
```

Here, the SELECT statement does not use tables and does not generate messages, so it does not change the diagnostics area. Consequently, SHOW WARNINGS output pertains to the DROP TABLE statement.

Now:

```
mysql> DROP TABLE test.no_such_table;
ERROR 1051 (42S02): Unknown table 'test.no_such_table'
mysql> SELECT @@warning_count;
Query OK, 0 rows affected (0.00 sec)
mysql> SHOW WARNINGS;
Empty set (0.00 sec)
```

Here, the SELECT statement clears the diagnostics area because it is a nondiagnostic statement. Consequently, SHOW WARNINGS output pertains to the SELECT statement (and is empty because the SELECT produces no messages).

An implication of this change in diagnostics area handling is that if you expect to display the warning count as well as the list of messages, you should list the messages first because selecting the warning_count value clears the message list. Alternatively, use SHOW COUNT(*) WARNINGS to display the count; this is recognized as a diagnostic statement and does not clear the diagnostics area. Similar considerations apply to use of error_count.

For compliance with the SQL standard, which states that diagnostics statements are not preparable, MySQL no longer supports the following as prepared statements:

- SHOW WARNINGS, SHOW COUNT(*) WARNINGS
- SHOW ERRORS, SHOW COUNT(*) ERRORS
- Statements containing any reference to the warning_count or error_count system variable.

In other words, those statements are now treated, in terms of preparability, the same as GET DIAGNOSTICS, which was already not preparable.

Logging Notes

- **Incompatible Change:** Several changes have been made to provide more logging control and more informative log messages:

  - The `log_error_verbosity` system variable now controls verbosity of the server in writing error, warning, and note messages to the error log. Permitted values are 1 (errors only), 2 (errors and warnings), 3 (errors, warnings, and notes), with a default of 3.

    `log_error_verbosity` is preferred over, and should be used instead of, the older `log_warnings` system variable. See the description of `log_warnings` for information about how that variable relates to `log_error_verbosity` (Server System Variables). The `log_warnings` system variable and `--log-warnings` command-line option now are deprecated and will be removed in a future MySQL release.

    Note

    The effective default verbosity is different now. The previous default (`log_warnings=1`) corresponds to `log_error_verbosity=2`, but the default `log_error_verbosity` is 3. To achieve a logging level similar to the previous default, set `log_error_verbosity=2`.

  - Default server verbosity is less when invoked with the `--bootstrap` option (such as is done by `mysql_install_db`): Only errors are written during the installation process so that they are less likely to be overlooked by the installer.

  - The `log_timestamps` system variable has been introduced for control of the timestamp time zone of messages written to the error log, and of general query log and slow query log messages written to files. (It does not affect the time zone of general query log and slow query log messages written to log tables, but rows retrieved from those tables can be converted from the local system time zone to any desired time zone with `CONVERT_TZ()` or by setting the session `time_zone` system variable.)

    Note

    The default timestamp time zone is different now (`UTC` rather than the local system time zone). To restore the previous default, set `log_timestamps=SYSTEM`.

  - The format of timestamps has changed for messages written to the error log, and for general query log and slow query log messages written to files. Timestamps are written using ISO 8601 / RFC 3339 format: `YYYY-MM-DDThh:mm:ss.uuuuuu` plus a tail value of `Z` signifying Zulu time (UTC) or `±hh:mm` (an offset from UTC). In addition, for the general query log file, timestamps are included in every message, not just when the second changes.

    The format of timestamps has also changed for messages written to the general query log and slow query log tables (`mysql.general_log`, `mysql.slow_log`), which now include fractional seconds. (The column type for timestamps has changed from `TIMESTAMP` to `TIMESTAMP(6)`.)

  - Previously, the ID included in error log messages was the `mysqld` process ID. Now the ID is that of the thread within `mysqld` responsible for writing the message. This is more informative with respect to which part of the server produced the message. It is also more consistent with general query log and slow query log messages, which include the connection thread ID.

  For information about log output destinations, see Selecting General Query and Slow Query Log Output Destinations. For information about specific logs, see The Error Log, The General Query Log, and The Slow Query Log.

Performance Schema Notes

- The Performance Schema now instruments memory usage and aggregates memory usage statistics, detailed by these factors:

  - Type of memory used (various caches, internal buffers, and so forth)

  - Thread, account, user, host indirectly performing the memory operation

  The Performance Schema instruments the following aspects of memory use

  - Memory sizes used

  - Operation counts

  - Low and high water marks

  Memory sizes help to understand or tune the memory consumption of a server.

  Operation counts help to understand or tune the overall pressure the server is putting on the memory allocator, which has an impact on performance. Allocating a single byte one million times is not the same as allocating one million bytes a single time; tracking both sizes and counts can expose the difference.

  Low and high water marks are critical to detect workload spikes, overall workload stability, and possible memory leaks.

  These specific changes were implemented:

  - The `setup_instruments` table now has memory instruments. These have names of the form `memory/`*`component/instrument_name`*. Memory instrumentation is disabled by default.

  - The `performance_schema_max_memory_classes` system variable configures the maximum number of memory instruments.

  - The `Performance_schema_memory_classes_lost` status variable indicates the number of times a memory instrument could not be loaded.

  - Several summary tables aggregate memory-related events.

  For more information, see Memory Summary Tables.

  If you upgrade to this release of MySQL from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `performance_schema` database.

- The Performance Schema now instruments stored program execution and aggregates statistics for them. This includes stored procedures, stored functions, triggers, and Event Scheduler events.

  These specific changes were implemented:

  - The `setup_instruments` table has new instruments. The `statement/scheduler/event` instrument tracks all events executed by the Event Scheduler. Instruments with names of the form `statement/sp/`*`program_instruction`* track internal instructions executed by stored programs.

  - The `setup_objects` table `OBJECT_TYPE` column now permits values of `'EVENT'`, `'FUNCTION'`, `'PROCEDURE'`, `'TABLE'`, or `'TRIGGER'`, not just `'TABLE'`.

  - Statement event tables (events_statements_current, events_statements_history, and events_statements_history_long) now have a NESTING_LEVEL column that indicates the event nesting level.

  - The performance_schema_max_program_instances and performance_schema_max_statement_stack system variables configure the maximum number of stored programs and the maximum depth of nested stored program calls for which the Performance Schema maintains statistics.

  - The Performance_schema_program_lost and Performance_schema_nested_statement_lost status variables indicate the number of stored programs for which statistics were lost, and the number of stored program statements for which statistics were lost.

  - The events_statements_summary_by_program summary table aggregates statement events per stored program.

  For more information, see Event Pre-Filtering, and Statement Summary Tables.

  If you upgrade to this release of MySQL from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate these changes into the performance_schema database.

- The Performance Schema now provides tables that expose replication information. This is similar to the information available from the SHOW SLAVE STATUS statement, but representation in table form is more accessible and has usability benefits:

  - SHOW SLAVE STATUS output is useful for visual inspection, but not so much for programmatic use. By contrast, using the Performance Schema tables, information about slave status can be searched using general SELECT queries, including complex WHERE conditions, joins, and so forth.

  - Query results can be saved in tables for further analysis, or assigned to variables and thus used in stored procedures.

  - The replication tables provide better diagnostic information. For multi-threaded slave operation, SHOW SLAVE STATUS reports all coordinator and worker thread errors using the Last_SQL_Errno and Last_SQL_Error fields, so only the most recent of those errors is visible and information can be lost. The replication tables store errors on a per-thread basis without loss of information.

  - The last seen transaction is visible in the replication tables on a per-worker basis. This is information not avilable from SHOW SLAVE STATUS.

  - Developers familiar with the Performance Schema interface can extend the replication tables to provide additional information by adding rows to the tables.

  These tables provide replication information:

  - replication_connection_configuration and replication_connection_status indicate the configuration parameters used by the slave for connecting to the master and the status of the connection.

  - replication_execute_configuration and replication_execute_status indicate, for aspects of transaction execution on the slave not specific to any given thread, the configuration parameters and the current execution status.

- `replication_execute_status_by_coordinator` and `replication_execute_status_by_worker` contain thread-specific transaction execution information, either about the SQL thread (for a single-threaded slave), or about the coordinator and worker threads (for a multi-threaded slave).

If the slave is multi-threaded, the SQL thread is the coordinator for worker threads. In this case, the `Last_SQL_Error` field of `SHOW SLAVE STATUS` output now shows exactly what the `Last_Error_Message` column in the Performance Schema `replication_execute_status_by_coordinator` table shows. The field value is modified to suggest that there may be more failures in the other worker threads which can be seen in the `replication_execute_status_by_worker` table that shows each worker thread's status.

For more information, see Performance Schema Replication Tables.

If you upgrade to this release of MySQL from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `performance_schema` database.

RPM Notes

- It was not possible to upgrade a community RPM to a commercial RPM using `rpm -uvh` or `yum localupdate`. To deal with this, the RPM spec file has been updated in MySQL 5.7.2, which has the following consequences:

  - For a non-upgrade installation (no existing MySQL version installed), it possible to install MySQL using `yum`.

  - For upgrades, it is necessary to clean up any earlier MySQL installations. In effect, the update is performed by removing the old installations and installing the new one.

  Additional details follow.

  For a non-upgrade installation of MySQL 5.7.2, it is possible to install using `yum`:

  ```
  shell> yum install MySQL-server-NEWVERSION.glibc23.i386.rpm
  ```

  For upgrades to MySQL 5.7.2, the upgrade is performed by removing the old installation and installing the new one. To do this, use the following procedure:

  1. Remove the existing 5.7.*X* installation. *`OLDVERSION`* is the version to remove.

     ```
     shell> rpm -e MySQL-server-OLDVERSION.glibc23.i386.rpm
     ```

     Repeat this step for all installed MySQL RPMs.

  2. Install the new version. *`NEWVERSION`* is the version to install.

     ```
     shell> rpm -ivh MySQL-server-NEWVERSION.glibc23.i386.rpm
     ```

  Alternatively, the removal and installation can be done using `yum`:

  ```
  shell> yum remove MySQL-server-OLDVERSION.glibc23.i386.rpm
  shell> yum install MySQL-server-NEWVERSION.glibc23.i386.rpm
  ```

  (Bug #16445097, Bug #16445125, Bug #16587285)

Security Notes

- Platform availability, usability, and security of mysql_secure_installation has been improved. Previously, this program was a shell script available for Unix and Unix-like systems. It has been converted to a binary executable program (written in C++) that is available on all platforms. Implementation as a C++ program permits mysql_secure_installation to connect directly to the MySQL server using the client/server protocol, rather than by invoking mysql to do so and communicating with mysql using temporary files.

  This reimplementation of mysql_secure_installation is feature-compatible with previous versions, but the following usability improvements have been made:

  - The validate_password plugin can be used for password strength checking.

  - Standard MySQL options such as --host and --port are supported on the command line and in option files.

  For more information, see mysql_secure_installation — Improve MySQL Installation Security. For more information about validate_password, see The Password Validation Plugin.

Semisynchronous Replication Notes

- **Replication:** Semisynchronous replication master servers now use a different wait point by default in communicating wih slaves. This is the point at which the master waits for acknowledgment of transaction receipt by a slave before returning a status to the client that committed the transaction. The wait point is controlled by the new rpl_semi_sync_master_wait_point system variable. These values are permitted:

  - AFTER_SYNC (the default): The master writes each transaction to its binary log and the slave, and syncs the binary log to disk. The master waits for slave acknowledgment of transaction receipt after the sync. Upon receiving acknowledgment, the master commits the transaction to the storage engine and returns a result to the client, which then can proceed.

  - AFTER_COMMIT: The master writes each transaction to its binary log and the slave, syncs the binary log, and commits the transaction to the storage engine. The master waits for slave acknowledgment of transaction receipt after the commit. Upon receiving acknowledgment, the master returns a result to the client, which then can proceed.

    For older versions of MySQL, semisynchronous master behavior is equivalent to a setting of AFTER_COMMIT.

  The replication characteristics of these settings differ as follows:

  - With AFTER_SYNC, all clients see the committed transaction at the same time: After it has been acknowledged by the slave and committed to the storage engine on the master. Thus, all clients see the same data on the master.

    In the event of master failure, all transactions committed on the master have been replicated to the slave (saved to its relay log). A crash of the master and failover to the slave is lossless because the slave is up to date.

  - With AFTER_COMMIT, the client issuing the transaction gets a return status only after the server commits to the storage engine and receives slave acknowledgment. After the commit and before slave acknowledgment, other clients can see the committed transaction before the committing client.

If something goes wrong such that the slave does not process the transaction, then in the event of a master crash and failover to the slave, it is possible that such clients will see a loss of data relative to what they saw on the master.

The new wait point is a behavior change, but requires no reconfiguration. The change does introduce a version compatibility constraint because it increments the semisynchronous interface version: Servers for MySQL 5.7.2 and up do not work with semisynchronous replication plugins from older versions, nor do servers from older versions work with semisynchronous replication plugins for MySQL 5.7.2 and up.

Trigger Notes

- Previously, a table could have at most one trigger for each combination of trigger event (`INSERT`, `UPDATE`, `DELETE`) and action time (`BEFORE`, `AFTER`). This limitation has been lifted and multiple triggers are permitted. Along with that change, several additional modifications were made:

  - By default, triggers for each combination of trigger event and action time execute in the order they were created. To make it possible to specify trigger activation order, `CREATE TRIGGER` now supports `FOLLOWS` and `PRECEDES` clauses. Each clause takes the name of an existing trigger that has the same trigger event and action time.

  - The `ACTION_ORDER` column in the `INFORMATION_SCHEMA.TRIGGERS` table is no longer 0 but an integer greater than zero that indicates the order in which triggers activate.

  - Creation time for triggers is now maintained, as a `TIMESTAMP(2)` value (with a fractional part in hundredths of seconds):

    - The `CREATED` column in the `TRIGGERS` table is no longer `NULL`, for triggers created as of MySQL 5.7.2.

    - The same is true for the `Created` column of `SHOW TRIGGERS` output, and for the (new) `Created` column of `SHOW CREATE TRIGGER` output.

    - The *`tbl_name`*`.TRG` file that stores trigger information for table *`tbl_name`* now contains a `created` line with trigger creation times.

    For additional information, see Using Triggers, `CREATE TRIGGER` Syntax, `SHOW CREATE TRIGGER` Syntax, `SHOW TRIGGERS` Syntax, The `INFORMATION_SCHEMA TRIGGERS` Table, and Table Trigger Storage.

  - If run against a table that has triggers, `mysql_upgrade` and `CHECK TABLE ... FOR UPGRADE` display this warning for each trigger created before MySQL 5.7.2:

    ```
    Trigger db_name.tbl_name.trigger_name does not have CREATED attribute.
    ```

    The warning is informational only. No change is made to the trigger.

  These changes have implications for backups, upgrades, and downgrades, as described following. For brevity, “multiple triggers” here is shorthand for “multiple triggers that have the same trigger event and action time.”

  Backup and restore. `mysqldump` dumps triggers in activation order so that when the dump file is reloaded, triggers are re-created in the same activation order. However, if a `mysqldump` dump file contains multiple triggers for a table that have the same trigger event and action time, an error occurs for attempts to load the dump file into an older server that does not support multiple triggers. (See the downgrading notes for a workaround; you can convert triggers to be compatible with older servers.)

Upgrades. Suppose that you upgrade an old server that does not support multiple triggers to MySQL 5.7.2 or newer. If the new server is a replication master and has old slaves that do not support multiple triggers, an error occurs on those slaves if a trigger is created on the master for a table that already has a trigger with the same trigger event and action time. To avoid this problem, upgrade the slaves first, then upgrade the master.

Downgrades. If you downgrade a server that supports multiple triggers to an older version that does not, the downgrade has these effects:

- For each table that has triggers, all trigger definitions remain in the `.TRG` file for the table. However, if there are multiple triggers with the same trigger event and action time, the server executes only one of them when the trigger event occurs. For information about `.TRG` files, see Table Trigger Storage.
- If triggers for the table are added or dropped subsequent to the downgrade, the server rewrites the table's `.TRG` file. The rewritten file retains only one trigger per combination of trigger event and action time; the others are lost.

To avoid these problems, modify your triggers before downgrading. For each table that has multiple triggers per combination of trigger event and action time, convert each such set of triggers to a single trigger as follows:

1. For each trigger, create a stored routine that contains all the code in the trigger. Values accessed using `NEW` and `OLD` can be passed to the routine using parameters. If the trigger needs a single result value from the code, you can put the code in a stored function and have the function return the value. If the trigger needs multiple result values from the code, you can put the code in a stored procedure and return the values using `OUT` parameters.
2. Drop all triggers for the table.
3. Create one new trigger for the table that invokes the stored routines just created. The effect for this trigger is thus the same as the multiple triggers it replaces.

Functionality Added or Changed

- **Performance; InnoDB:** Memory for transaction instances (`trx_t`) is now allocated in configurable sized blocks that are a multiple of transaction instance size. Transaction instances are also placed in a priority queue and ordered by their address in memory so that when instances are allocated from the pool, they are close together. This enhancement reduces the cost incurred by iterating over transactions instances when allocating instances from the pool.
- **Performance; InnoDB:** Internal B-tree index operations have been optimized to reduce index locking contention.
- **Performance; InnoDB:** Multi-version concurrency control (MVCC) in `InnoDB` requires that each transaction using MVCC be assigned a read view. To improve `InnoDB` read-only and read-write performance, read view creation has been optimized by reducing mutex contention.
- **Incompatible Change:** Previously, the Performance Schema statement instrumentation did not include statements executed on a slave replication server. To address this, a new abstract instrument, `statement/rpl/relay_log`, has been added to the `setup_instruments` table. This instrument is used during the early stages of replicated statement classification before the exact statement type is known. (Bug #16750433, Bug #17271055)
- **Incompatible Change:** Previously, the main loop responsible for accepting client connections also performed initialization of data structures related to each connection. These initialization tasks now

are delegated to worker threads to minimize work done by the accept loop and maximize connection acceptance rate.

As a result of this change, the bind_address, thread_cache_size, and thread_handling system variables are no longer visible to the embedded server (libmysqld). Similarly, the Slow_launch_threads and Threads_cached status variables are not meaningful in the embedded server. These variables are no longer visible within the embedded server and embedded applications that rely on these variables should be modified accordingly. (Bug #62288, Bug #12951536, Bug #62284, Bug #12951595, Bug #62283, Bug #12951605)

- **Incompatible Change:** The unused --basedir and --datadir options for mysql_upgrade have been removed.

- **Important Change; Partitioning:** It is now possible to check and to repair partitions which contain duplicate key violations. This is implemented by allowing the IGNORE keyword in ALTER TABLE statements using the CHECK PARTITION or REPAIR PARTITION option. The keyword has the following effects on the behavior of these statements:

  - ALTER IGNORE TABLE ... REPAIR PARITION removes all rows that cannot be moved due to the presence of duplicate keys.

  - ALTER IGNORE TABLE ... CHECK PARTITION reports the contents of all columns in the partitioning expression for each row.

  > **Note**
  >
  > Support for the IGNORE keyword is removed in MySQL 5.7.4.

  (Bug #16900947)

- **Important Change; Replication:** By default, when promoting integers from a smaller type on the master to a larger type on the slave (for example, from a SMALLINT column on the master to a BIGINT column on the slave), the promoted values are treated as though they are signed. Now in such cases it is possible to modify or override this behavior using one or both of ALL_SIGNED, ALL_UNSIGNED in the set of values specified for the slave_type_conversions server system variable. For more information, see Row-based replication: attribute promotion and demotion, as well as the description of the variable. (Bug #15831300)

- **InnoDB:** innochecksum functionality has been enhanced with new options and extended capabilities. See innochecksum — Offline InnoDB File Checksum Utility. (Bug #16945722)

- **InnoDB:** A new CMake option, WITH_INNODB_EXTRA_DEBUG, has been added that enables additional InnoDB debug checks. WITH_INNODB_EXTRA_DEBUG can only be enabled when the WITH_DEBUG option is also enabled. (Bug #16821155)

- **InnoDB:** A number of internal debug flags in the InnoDB code could only be set at compilation time or from a debugger. As a result, a significant amount of diagnostic information was unused. This enhancement replaces internal debug flags with DBUG labels so that the DBUG package can be used and printouts from various InnoDB subsystems can be enabled using the mysqld --debug command line option. See the Debugging a MySQL Server section for information about configuring MySQL for debugging, creating trace files, and using the mysqld --debug option.

- **InnoDB:** The process for converting a transaction's implicit lock to an explicit lock has been optimized to improve performance. The optimization reduces lock_sys_t::mutex contention.

- **InnoDB:** Beginning with MySQL 5.7.2, UPDATE_TIME displays a timestamp value for the last UPDATE, INSERT, or DELETE performed on InnoDB tables. Previously, UPDATE_TIME displayed a NULL value for InnoDB tables. For MVCC, the timestamp value reflects the COMMIT time, which is considered the last update time. Timestamps are not persisted when the server is restarted or when the table is evicted from the InnoDB data dictionary cache.

- **InnoDB:** For SELECT COUNT (*) queries, where a table's committed record count is changed by transaction deltas, there is now a single handler call to the storage engine to return the record count to the optimizer instead of one call for each record. This change generally improves SELECT COUNT (*) query performance and reduces in-memory table scan cost, as each record is no longer returned to the optimizer.

  In some instances, however, where there is a large clustered index and a very small secondary index, performance may not be improved. Previously, the optimizer would choose to traverse the smaller secondary index instead of the larger clustered index. The smaller secondary index could, in this case, offer better performance than a clustered index with a single handler call to the storage engine. However, there may be no performance benefit if the secondary index is often updated. When a secondary index page is modified by a transaction that is more recent than the COUNT(*) transaction, InnoDB must read the clustered index to determine if the record is visible. In this case, InnoDB would read both the secondary and clustered index, which is costlier than reading only the clustered index.

- **InnoDB:** Read-only transactions will no longer be assigned a transaction ID. Conversely, an ID will only be assigned if a transaction is explicitly tagged as “read-write”, if a transaction has acquired an X or IX lock on a table, or if a transaction is a read-only transaction writing to a temporary table. All other transactions are considered “read-only” and are not assigned an ID. Furthermore, read-only transactions are not tagged as “read-only” unless they are explicitly started with START TRANSACTION READ ONLY. For transactions without transaction IDs, SHOW ENGINE INNODB STATUS prints an identifier that is unique but only within the context of the SHOW ENGINE INNODB STATUS invocation.

- **InnoDB:** MySQL 5.7.2 introduces a new type of undo log for both normal and compressed temporary tables and related objects. The new type of undo log is not a redo log, as temporary tables are not recovered during crash recovery and do not require redo logs. The new undo log resides in the temporary tablespace. The default temporary tablespace file, ibtmp1, is located in the data directory by default and is always recreated on server startup. A user defined location for the temporary tablespace file can be specified by setting innodb_temp_data_file_path. For more information, see InnoDB Temporary Table Undo Logs.

- **InnoDB:** The limit on concurrent data-modifying transactions is now 96 * 1023 transactions that generate undo records. As of MySQL 5.7.2, 32 of 128 rollback segments are allocated to non-redo logs for transactions that modify temporary tables and related objects. This reduces the maximum number of concurrent data-modifying transactions from 128K to 96K. The 96K limit assumes that transactions do not modify temporary tables. If all data-modifying transactions also modify temporary tables, the limit would be 32K concurrent transactions.

- **InnoDB:** DML operations (INSERT, UPDATE, DELETE) for temporary tables have been optimized by turning off redo logging, locking, and change buffering that is not required for temporary tables. Turning off these functions optimizes temporary table DML operations by reducing associated I/O.

- **InnoDB:** InnoDB buffer pool dump and load operations have been enhanced. A new system variable, innodb_buffer_pool_dump_pct, allows you to specify the percentage of most recently used pages in each buffer pool to read out and dump. When there is other I/O activity being performed by InnoDB background tasks, InnoDB attempts to limit the number of buffer pool load operations per second using the innodb_io_capacity setting.

- **InnoDB:** Buffer pool list scans and related batch processing have been optimized to reduce scan complexity and the number of pages scanned.

- **InnoDB:** Refactored mutex code makes selecting the appropriate mutex easier and allows multiple mutex types to be combined in the same instance. The refactored code also removes the distinction between `fast_mutex_t` and home brew `ib_mutex_t` types, implements a common interface for both mutex types, and allows new mutex types to be added in the future. Additionally, mutex code is decoupled from `InnoDB` code so that it can be used as a library, and a “PolicyMutex” interface has been introduced. The new interface uses static inheritance (templates) for mutex implementation making it easier to define policies and customize mutex behavior.

- **Partitioning:** The following operations are now supported for individual subpartitions as well as partitions: `ANALYZE`, `CHECK`, `OPTIMIZE`, `REBUILD`, `REPAIR`, and `TRUNCATE` (see `ALTER TABLE` Partition Operations). (Bug #14028340, Bug #65184)

- **Replication:** The master dump thread was refactored to reduce lock contention and improve master throughput. Previously, the dump thread took a lock on the binary log whenever reading an event; now the lock is held only while reading the position at the end of the last successfully written event. This means that multiple dump threads can now read concurrently from the binary log file, and that dump threads can read while clients write to the binary log.

- **Replication:** Previously, transactions could be applied in parallel only if they did not touch the same database. However, the MySQL Server uses a lock-based scheduler, which means that it should be possible to execute in parallel all uncommitted replication threads already in the prepare phase, without violating consistency. Such parallel execution can now be enabled on the slave by starting the slave `mysqld` with `--slave-parallel-type=LOGICAL_CLOCK` or, if `mysqld` is already started, by setting the value of the global system variable `slave_parallel_type` to `'LOGICAL_CLOCK'` on a stopped slave.

  When this feature is enabled, each transaction is marked with a logical timestamp. This timestamp identifies the last transaction committed at the time that the current transaction entered the prepare stage, and all transactions having the same timestamp can execute in parallel.

  To disable this feature without restarting, stop the slave using `STOP SLAVE` (if it is running as a slave), issue `SET @global-slave_parallel_type='DATABASE'`, then issue `START SLAVE` when you want the slave to resume. You can also disable the feature by restarting the slave `mysqld` without the `--slave-parallel-type` option, or by setting it explicitly to `DATABASE`. When parallel execution of preapred transactions is disabled, the slave follows the old behavior and applies in parallel only those transactions that do not cause changes in the same database.

- Support for LinuxThreads has been removed from the source code. LinuxThreads was superseded by NPTL in Linux 2.6. (Bug #17007529)

- Previously, program options could be specified in full or as any unambiguous prefix. For example, the `--compress` option could be given to `mysqldump` as `--compr`, but not as `--comp` because the latter is ambiguous. Option prefixes are no longer supported; only full options are accepted. This is because prefixes can cause problems when new options are implemented for programs and a prefix that is currently unambiguous might become ambiguous in the future. (Bug #16996656)

- Support for building Apple universal binaries to support PowerPC has been removed from the source code. (Bug #16959103)

- `CMake` no longer checks for `memmove()` or `memcpy()` because they are standard C library functions. Also, implementation of the `bmove_upp()` function was replaced with calls to `memmove()`, which may have positive performance implications. (Bug #16839824)

- The C API `libmysqlclient` shared-library `.so` files now have version 18.1.0 (up from version 18.0.0 used in MySQL 5.5). 18.1.0 can be used as a replacement for 18.0.0. (Bug #16809055, Bug #59106, Bug #12407476)

- Use of DYNAMIC_ARRAY was reduced, which improves performance of certain range queries by 3-4%. (Bug #16736776, Bug #17030235)

- mysql_upgrade now verifies that the server version matches the version against which it was compiled, and exits if there is a mismatch. In addiion, a --version-check option permits specifying whether to enable version checking (the default), or disable checking if given as --skip-version-checking. (Bug #16500013)

- mysqladmin now supports a --show-warnings option to display warnings resulting from execution of statements sent to the server. (Bug #16517756)

- The following items are deprecated and will be removed in a future MySQL release. Where alternatives are shown, applications should be updated to use them.

  - The ENCODE() and DECODE() functions. Use AES_ENCRYPT() and AES_DECRYPT() instead.

  - The INFORMATION_SCHEMA.PROFILING table. Use the Performance Schema instead; see MySQL Performance Schema.

  (Bug #16463921)

- Invoking CMake with -DWITH_AUTHENTICATION_PAM=1 now causes the build to fail (rather than issue only a warning) if the PAM plugin cannot be built. (Bug #14554639)

- In batch mode, mysql formatted result status messages such as ""Query OK, 1 row affected"" but did not print them. Now these messages are not formatted. (Bug #69486, Bug #16971432)

- Several inefficiencies were corrected:

  - A loop in Item_in_subselect::single_value_transformer() could execute too many times.

  - The myisamchk(), my_test_if_sort_rep(), and recreate_table() functions in MyISAM code could execute too many times.

  Thanks to Po-Chun Chang for the patches to correct these issues. (Bug #69138, Bug #16764131, Bug #69117, Bug #16751784, Bug #69561, Bug #17007268, Bug #69553, Bug #17001703)

- Plugins can now define and expose floating-point system variables of type double using the MYSQL_SYSVAR_DOUBLE() and MYSQL_THDVAR_DOUBLE() accessor macros. See Client Plugin Descriptors. (Bug #68121, Bug #16194302)

- Semi-join LooseScan strategy now can use ref access and applies to a wider range of queries.

- EXPLAIN can now be used to obtain the execution plan for an explainable statement executing in a named connection:

  ```
  EXPLAIN [options] FOR CONNECTION connection_id;
  ```

  For example, if you are running a statement in one session that is taking a long time to complete, using EXPLAIN FOR CONNECTION in another session may yield useful information about the cause of the delay and help you optimize the statement.

  connection_id is the connection identifier, as obtained from the INFORMATION_SCHEMA PROCESSLIST table or the SHOW PROCESSLIST statement. If you have the PROCESS privilege, you can specify the identifier for any connection. Otherwise, you can specify the identifier only for your own connections.

  Changes in EXPLAIN output:

  - In the output from EXPLAIN FOR CONNECTION, an Extra value of Plan isn't ready yet means that the optimizer has not finished creating the execution plan for the statement executing in the named connection. (For JSON-format output, this is indicated by planned: false.)

  - In the output from any EXPLAIN used to obtain the execution plan for non-SELECT statements, the select_type value displays the statement type for affected tables. For example, select_type is DELETE for DELETE statements.

  For more information, see EXPLAIN Syntax.

- To make it easier to see the difference between good and bad execution plans, JSON-format EXPLAIN output now includes this additional cost information:

  - query_cost: The total cost of a query block, whether a top-level query or subquery. For a top-level SELECT, this should be equal to the Last_query_cost status variable.

  - sort_cost: The cost of the first sorting operation (GROUP BY or ORDER BY) where and if filesort is used.

  - read_cost: The cost of reading data from each table used in the query block (that is, access method cost).

  - eval_cost: The cost of condition evaluation for each table in the query block.

  - prefix_cost: The cost of executing prefix join in the query block; that is, the cost of joining tables of the query block from the first one to the one (and including it) for which the value is given.

  - data_read_per_join: The estimated amount of data processed by the handler interface per query or subquery execution. This is essentially record width * number of read records.

  - rows_produced_per_join/ rows_examined_per_join: The estimated number of records from the table (per table from the query block) produced or examined per single query block execution.

  - used_columns: The list of columns from the table (per each table in the query block) used for either read or write in the query.

  This cost information is not displayed for INFORMATION_SCHEMA tables.

- MySQL now supports the use of protocol trace plugins: client-side plugins that implement tracing of communication between a client and the server that takes place using the client/server protocol. Protocol trace plugins use the client plugin API.

  In MySQL source distributions, a test protocol trace plugin is implemented in the test_trace_plugin.cc file in the libmysql directory. This can be examined as a guide to writing other protocol trace plugins.

  For more information, see Writing Plugins; in particular, Writing Protocol Trace Plugins.

Bugs Fixed

- **Performance; Important Change; InnoDB:** InnoDB would fail to open a tablespace that has multiple data files. This removes the known limitation that was in MySQL Server 5.6.12. (Bug #17033706, Bug #69623)

- **Performance; InnoDB:** A code regression introduced in MySQL 5.6 negatively impacted DROP TABLE and ALTER TABLE performance. This could cause a performance drop between MySQL Server 5.5.x and 5.6.x. (Bug #16864741, Bug #69316)

- **Performance; InnoDB:** When innodb_thread_concurrency is set to a non-zero value, there was a possibility that all innodb_concurrency_tickets would be released after each row was read, resulting in a concurrency check after each read. This could impact performance of all queries. One symptom could be higher system CPU usage. We strongly recommend that you upgrade to MySQL Server 5.6.13 if you use this setting. This could cause a performance drop between MySQL Server 5.5.x and 5.6.x. (Bug #68869, Bug #16622478)

- **Incompatible Change:** When used for an existing MySQL account, the GRANT statement could produce unexpected results if it included an IDENTIFIED WITH clause that named an authentication plug differing from the plugin named in the corresponding mysql.user table row.

  Because IDENTIFIED WITH is intended only for GRANT statements that create a new user, it is now prohibited if the named account already exists. (Bug #16083276)

- **Incompatible Change:** It is possible for a column DEFAULT value to be valid for the sql_mode value at table-creation time but invalid for the sql_mode value when rows are inserted or updated. Example:

  ```
  SET sql_mode = '';
  CREATE TABLE t (d DATE DEFAULT 0);
  SET sql_mode = 'NO_ZERO_DATE,STRICT_ALL_TABLES';
  INSERT INTO t (d) VALUES(DEFAULT);
  ```

  In this case, 0 should be accepted for the CREATE TABLE but rejected for the INSERT. However, the server did not evaluate DEFAULT values used for inserts or updates against the current sql_mode. In the example, the INSERT succeeds and inserts '0000-00-00' into the DATE column.

  The server now applies the proper sql_mode checks to generate a warning or error at insert or update time.

  A resulting incompatibility for replication if you use statement-based logging (binlog_format=STATEMENT) is that if a slave is upgraded, a nonupgraded master will execute the preceding example without error, whereas the INSERT will fail on the slave and replication will stop.

  To deal with this, stop all new statements on the master and wait until the slaves catch up. Then upgrade the slaves followed by the master. Alternatively, if you cannot stop new statements, temporarily change to row-based logging on the master (binlog_format=ROW) and wait until all slaves have processed all binary logs produced up to the point of this change. Then upgrade the slaves followed by the master and change the master back to statement-based logging. (Bug #68041, Bug #16078943)

- **Important Change; Replication:** When the server was running with --binlog-ignore-db and SELECT DATABASE() returned NULL (that is, there was no currently selected database), statements using fully qualified table names in *dbname*.*tblname* format were not written to the binary log. This was because the lack of a currently selected database in such cases was treated as a match for any possible ignore option rather than for no such option; this meant that these statements were always ignored.

  Now, if there is no current database, a statement using fully qualified table names is always written to the binary log. (Bug #11829838, Bug #60188)

- **InnoDB; Partitioning:** Joins involving partitioned InnoDB tables having one or more BLOB columns were not always handled correctly. The BLOB column or columns were not required to be join columns, or otherwise to be named or referenced in the statement containing the join, for this issue to occur. (Bug #16367691)

- **InnoDB; Partitioning:** Following any query on the INFORMATION_SCHEMA.PARTITIONS table, InnoDB index statistics as shown in the output of statements such as SELECT * FROM

`INFORMATION_SCHEMA.STATISTICS` were read from the last partition, instead of from the partition containing the greatest number of rows. (Bug #11766851, Bug #60071)

References: See also Bug #16882435, Bug #69179.

- **InnoDB; Replication:** Trying to update a column, previously set to `NULL`, of an `InnoDB` table with no primary key caused replication to fail on the slave with `Can't find record in 'table'`.

  Note

  This issue was inadvertently reintroduced in MySQL 5.6.6, and fixed again in MySQL 5.6.12.

  (Bug #11766865, Bug #60091)

  References: See also Bug #16566658.

- **InnoDB:** When logging the delete-marking of a record during online `ALTER TABLE...ADD PRIMARY KEY`, `InnoDB` writes the transaction ID to the log as it was before the deletion or delete-marking of the record. When doing this, `InnoDB` would overwrite the `DB_TRX_ID` field in the original table, which could result in locking issues. (Bug #17316731)

- **InnoDB:** The `row_sel_sec_rec_is_for_clust_rec` function would incorrectly prepare to compare a NULL column prefix in a secondary index with a non-NULL column in a clustered index. (Bug #17312846)

- **InnoDB:** An incorrect purge would occur when rolling back an update to a delete-marked record. (Bug #17302896)

- **InnoDB:** An assertion failure would occur while searching an index tree and traversing multiple levels where a block is accessed or pinned at each level. (Bug #17315967)

- **InnoDB:** In Windows 64-bit debug builds, read view `COPY_TRX_IDS` would report a “`vector subscript out of range`” error to standard error output. (Bug #17320056)

- **InnoDB:** The assertion `ut_ad(oldest_lsn <= cur_lsn)` in file `buf0flu.cc` would fail because the current max LSN would be retrieved from the buffer pool before the oldest LSN. (Bug #17252421)

- **InnoDB:** `InnoDB` `memcached` `add` and `set` operations would perform more slowly than SQL `INSERT` operations. (Bug #17214191)

- **InnoDB:** As commented in `log0log.h`, `old_lsn` and `old_buf_free` should only be compiled when `UNIV_LOG_DEBUG` is enabled. (Bug #17160270, Bug #69724)

- **InnoDB:** Before dropping an index, a check is performed to ensure the index root page is free. If the index root page is free, dropping activity is avoided. A transaction would be initialized before the check is performed. If the check evaluated to true, the initialized transaction would be left in a dangling state. (Bug #17076822)

- **InnoDB:** `InnoDB` would rename a user-defined foreign key constraint containing the string “_ibfk_” in its name, resulting in a duplicate constraint. (Bug #17076737, Bug #69693, Bug #17076718, Bug #69707)

- **InnoDB:** When started in ready-only mode, `InnoDB` would assert on a `SAVEPOINT`. (Bug #17086428)

- **InnoDB:** In debug builds, the `trx_sys->rw_max_trx_id` variable would sometimes be reversed resulting in an inconsistent `CLUST_INDEX_SIZE` value. (Bug #17026780)

- **InnoDB:** An `InnoDB` monitor test would raise an assertion in `ha_innodb.cc` due to a mutex conflict. (Bug #17027249)

- **InnoDB:** A regression introduced in the fix for Bug #14606334 would cause crashes on startup during crash recovery. (Bug #16996584)
- **InnoDB:** Rolling back an INSERT after a failed BLOB write would result in an assertion failure. The assertion has been modified to allow NULL BLOB pointers if an error occurs during a BLOB write. (Bug #16971045)
- **InnoDB:** The ha_innobase::clone function would incorrectly assert that a thread cannot clone a table handler that is used by another thread, and that the original table handler and the cloned table handler must belong to the same transaction. The incorrect assertions have been removed. (Bug #17001980)
- **InnoDB:** To avoid namespace clashes, usage of 'using namespace std' has been removed from InnoDB. (Bug #16899560)
- **InnoDB:** Optimized explicit record locking routines. (Bug #16880127)
- **InnoDB:** The server would crash during a memcached set operation. The failure was due to a padded length value for a utf8 char column. During a memcached update operation, a field from an old tuple would be copied with a data length that was less than the padded utf8 char column value. This fix ensures that old tuples are not copied. Instead, a new tuple is created each time. (Bug #16875543)
- **InnoDB:** When CHECK TABLE found a secondary index that contained the wrong number of entries, it would report an error but not mark the index as corrupt. CHECK TABLE now marks the index as corrupt when this error is encountered, but only the index is marked as corrupt, not the table. As a result, only the index becomes unusable until it is dropped and rebuilt. The table is unaffected. (Bug #16914007)
- **InnoDB:** InnoDB would attempt to gather statistics on partially created indexes. (Bug #16907783)
- **InnoDB:** A full-text search using the IN BOOLEAN MODE modifier would result in an assertion failure. (Bug #16927092)

  References: This bug is a regression of Bug #16516193.
- **InnoDB:** SHOW ENGINE INNODB STATUS output referenced a thread in hex format (example: thread handle 0x880), whereas the same thread was referenced in the SHOW ENGINE INNODB STATUS transaction list in decimal format (example: thread id 2176). (Bug #16934269, Bug #69437)
- **InnoDB:** When dropping all indexes on a column with multiple indexes, InnoDB would fail to block a DROP INDEX operation when a foreign key constraint requires an index. (Bug #16896810)
- **InnoDB:** The two INFORMATION_SCHEMA tables for the InnoDB buffer pool could show an invalid page type for read-fixed blocks. This fix will show the unknown page type for blocks that are I/O-fixed for reading. (Bug #16859867)
- **InnoDB:** InnoDB record comparison functions have been simplified and optimized. (Bug #16852278)
- **InnoDB:** innochecksum would ignore the return value of fwrite which could result in an error or generate warnings and compilation errors when WITH_INNODB_EXTRA_DEBUG CMake is enabled. (Bug #16872677)
- **InnoDB:** An assertion in row0mysql.cc, which ensures that the dictionary operation lock is not taken recursively, would fail. (Bug #16862290)
- **InnoDB:** An assertion failure would occur in file row0log.cc on ROW_FORMAT=REDUNDANT tables that contained an unexpected but valid data directory flag. (Bug #16863098)
- **InnoDB:** Removed invalid compilation warning messages that appeared when compiling the InnoDB memcached plugin. (Bug #16816824)

- **InnoDB:** The page_zip_validate() debug function, which is enabled when UNIV_ZIP_DEBUG is defined at compilation time, invokes page_zip_decompress(), which in turn would update some compression statistics. This would cause some mysql-test-run tests to fail. (Bug #16759605)

- **InnoDB:** Valgrind testing returned memory leak errors which resulted from a regression introduced by the fix for Bug #11753153. The dict_create_add_foreign_to_dictionary function would call pars_info_create but failed to call pars_info_free. (Bug #16754901)

- **InnoDB:** In debug builds, an online ALTER TABLE operation that performed a full table copy would raise an assertion. The assertion was due to a race condition that would occur during BLOB retrieval, when applying the table modification log to any log block except for the very last one. This fix modifies row_log_table_apply_convert_mrec() to ensure that an index B-tree lock is acquired to protect the access to log->blobs and the BLOB page. (Bug #16774118)

- **InnoDB:** When the function trx_rollback_or_clean_recovered() rolls back or cleans up transactions during a crash recovery, it removes the trx objects from the trx_sys list without freeing up the memory used by those objects. To prevent a memory leak, this fix adds trx_free_for_background() calls to trx_rollback_resurrected(), the function that removes the trx objects. (Bug #16754776)

- **InnoDB:** During an insert buffer merge, InnoDB would invoke lock_rec_restore_from_page_infimum() on a potentially invalid record pointer. (Bug #16806366)

- **InnoDB:** This patch removes the UNIV_INTERN function, which was introduced in MySQL 5.1 to help replace static linking in InnoDB with the shared object plugin. UNIV_INTERN is no longer required. (Bug #16781511)

- **InnoDB:** The innodb_rwlock_x_spin_waits item in the INFORMATION_SCHEMA.INNODB_METRICS table would show the same value as the innodb_rwlock_x_os_waits item. (Bug #16798175)

- **InnoDB:** In debug builds, an assertion could occur in OPT_CHECK_ORDER_BY when using binary directly in a search string, as binary may include NULL bytes and other non-meaningful characters. This fix will remove non-meaningful characters before the search is run. (Bug #16766016)

- **InnoDB:** The trx_tables_locked counter in INFORMATION_SCHEMA.INNODB_TRX would not account for all tables with locks. (Bug #16793724)

- **InnoDB:** A missing comma in SHOW STATUS output would break MySQL Enterprise Monitor parsing. (Bug #16723686)

- **InnoDB:** After a clean shutdown, InnoDB does not check .ibd file headers at startup. As a result, in a crash recovery scenario, InnoDB could load a corrupted tablespace file. This fix implements consistency and status checks to avoid loading corrupted files. (Bug #16720368)

- **InnoDB:** A memory leak would occur in dict_check_tablespaces_and_store_max_id() when space_id is equal to zero. (Bug #16737332)

- **InnoDB:** Some characters in the identifier for a foreign key constraint are modified during table exports. (Bug #16722314, Bug #69062)

- **InnoDB:** The page_zip_validate() consistency check would fail after compressing a page, in page_zip_compress(). This problem was caused by page_zip_decompress(), which would fail to set heap_no correctly when a record contained no user data bytes. A record with no user data bytes occurs when, for example, a primary key is an empty string and all secondary index fields are NULL or an empty string. (Bug #16736929)

- **InnoDB:** This patch is a code cleanup which may provide a minor performance improvement when keys are not used on columns and when using the default `latin1_swedish_ci` collation. (Bug #16723431)

- **InnoDB:** A regression introduced with the fix for Bug #11762038 would cause `InnoDB` to raise an incorrect error message. The message stated that, "InnoDB cannot delete/update rows with cascading foreign key constraints that exceed max depth of 20". The error message would occur when killing connections reading from `InnoDB` tables that did not have foreign key constraints. (Bug #16710923)

- **InnoDB:** In debug builds, an assertion failure would occur if `innodb_log_group_home_dir` does not exist. Instead of an assertion, `InnoDB` now aborts with an error message if `innodb_log_group_home_dir` does not exist. (Bug #16691130, Bug #69000)

- **InnoDB:** Stale `InnoDB` memcached connections would result in a memory leak. (Bug #16707516, Bug #68530)

- **InnoDB:** An `INSERT` into a temporary table resulted in the following assert: `ASSERT ID > 0 IN TRX_WRITE_TRX_ID()`. This fix corrects conditions for moving a transaction from a read-only list to a read-write list when the server is running in read-only mode. (Bug #16660575)

- **InnoDB:** A race condition would occur between `ALTER TABLE ... ADD KEY` and `INSERT` statements, resulting in an "Unable to Purge a Record" error. (Bug #16628233)

- **InnoDB:** Shutting down and restarting InnoDB with `--innodb-force-recovery` set to 3 or greater (4, 5, or 6) and attempting to drop a table would result in a crash. With `innodb_force_recovery` mode set to 3 or greater DML operations should be blocked and DDL operations allowed. This fix ensures that DDL operations are allowed. (Bug #16631778)

- **InnoDB:** Very large `InnoDB` full-text search (FTS) results could consume an excessive amount of memory. This bug fix reduces memory consumption for FTS results and introduces a new configuration parameter, `innodb_ft_result_cache_limit`, which places a default size limit of 2000000000 bytes on the `InnoDB` FTS query result cache. `innodb_ft_result_cache_limit` has an unlimited maximum value and can be set dynamically. (Bug #16625973)

- **InnoDB:** An existing full-text index would become invalid after running `ALTER TABLE ADD FULLTEXT` due to an unsynchronized full-text cache. (Bug #16662990, Bug #17373659)

- **InnoDB:** When `ADD PRIMARY KEY` columns are reordered in an `ALTER TABLE` statement (for example: `ALTER TABLE t1 ADD PRIMARY KEY(a,b), CHANGE a a INT AFTER b`), the log apply for `UPDATE` operations would fail to find rows. (Bug #16586355)

- **InnoDB:** DML operations on compressed temporary tables would result in a Valgrind error in the buffer manager stack. (Bug #16593331)

- **InnoDB:** In debug builds, the `assert_trx_in_list()` assert would fail, causing a race condition. This fix removes the assert. The same assert is verified in the caller and existing checks are sufficient. (Bug #16567258)

- **InnoDB:** A code regression resulted in a record lock wait in a dictionary operation. A code modification made to avoid starting a transaction on a temporary table failed to reset the state back to `init` upon completion of the operation. If a transaction is started, the state is usually reset by `trx_commit`. To catch similar problems in the future, this fix adds asserts to `innobase_commit()`, `innobase_rollback()`, and `ha_innobase::update_thd()` that trigger when `trx->dict_operation` and `trx->dict_operation_lock_mode` are not set. (Bug #16575799)

- **InnoDB:** This fix addresses a race condition that would occur between the rollback of a recovered transaction and creation of a secondary index in a locked operation. The race condition would corrupt the secondary index. (Bug #16593427)

- **InnoDB:** The MySQL printf facility (my_vsnprintf) did not understand the Microsoft I32 and I64 integer format width specifiers, such as %I64u for printing a 64-bit unsigned integer. As a result, DBUG_PRINT could not be used with the InnoDB UINT64PF format, which is defined as %I64u on Windows. This fix replaces the non-standard “I64” and “I32” length modifiers on Windows with “ll” and “l” so that they will be understood by both my_snprintf() and ut_snprintf(). (Bug #16559119)
- **InnoDB:** ALTER TABLE operations on InnoDB tables that added a PRIMARY KEY using a column prefix could produce an incorrect result. (Bug #16544336)
- **InnoDB:** For ALTER TABLE operations on InnoDB tables that required a table-copying operation, other transactions on the table might fail during the copy. However, if such a transaction issued a partial rollback, the rollback was treated as a full rollback. (Bug #16544143)
- **InnoDB:** Under certain circumstances, LRU flushing would take a long time possibly affecting all flushing activity and causing a shutdown timeout. (Bug #16500209)
- **InnoDB:** The recv_writer thread would only start after all redo log scans finished. In the case of multiple redo log scans, accumulated redo records would be applied after each scan and before processing the next scan. The absence of the recv_writer thread to help with flushing would slow recovery or result in a server startup timeout. This fix ensures that the recv_writer thread starts before the first scan batch is processed. (Bug #16501172)
- **InnoDB:** During a transaction commit, prepare_commit_mutex is acquired to preserve the commit order. If the commit operation failed, the transaction would be rolled back but the mutex would not be released. Subsequent insert operations would not be able to acquire the same mutex. This fix frees prepare_commit_mutex during innobase_rollback. (Bug #16513588)
- **InnoDB:** The row0purge.h include file contained a self-referencing inclusion. (Bug #16521741)
- **InnoDB:** The InnoDB memcached test.demo_test table would fail to work when defined as a utf8 charset table. (Bug #16499038)
- **InnoDB:** This fix replaces the IB_ULONGLONG_MAX constant with LSN_MAX where the code refers to log sequence numbers, or with TRX_ID_MAX where trx->no is initialized to an undefined value. This change does not alter the value of the constant. (Bug #16458660)
- **InnoDB:** This fix corrects the text for InnoDB error 6025, which stated, “InnoDB: read can't be opened in ./ib_logfile0 mode.”. The corrected message states, “InnoDB: ./ib_logfile0 can't be opened in read mode.” The variable and mode in the message construct were transposed. (Bug #16434398)
- **InnoDB:** The page_zip_available function would count some fields twice. (Bug #16463505)
- **InnoDB:** This fix removes most calls to OS_THREAD_SLEEP from InnoDB. (Bug #16472953, Bug #68588)
- **InnoDB:** Concurrently inserting into a full-text table would cause some inserts to fail. Duplicate values would be generated for full-text search document IDs when performing inserts into a hidden full-text search document ID column. (Bug #16469399)
- **InnoDB:** FLUSH TABLES FOR EXPORT would sleep too often while flushing pages from buffer pools. (Bug #16471701)
- **InnoDB:** In cases where threads are forced to do single page flushing, fsync() would be triggered for all data files. This fix allows for synchronous single page flushing. (Bug #16477781)
- **InnoDB:** When changing the shared tablespace file name using innodb_data_file_path and leaving the current log files in place, InnoDB would create a new tablespace file and overwrite the log files resulting in a mismatch between the data dictionary and tables on disk. This bug fix ensures

that InnoDB does not create a new tablespace if there are inconsistent system tablespaces, undo tablespaces, or redo log files. (Bug #16418661)

- **InnoDB:** An InnoDB memcached file descriptor leak would cause a serious error. (Bug #16466664)
- **InnoDB:** In debug builds, an insert would fail with an invalid assertion: sync_thread_levels_g(array, level - 1, TRUE). (Bug #16409715)
- **InnoDB:** Multiple concurrent calls to dict_update_statistics() would result in unnecessary server load. (Bug #16400412)
- **InnoDB:** On 64-bit Windows builds, INNODB_BUFFER_POOL_SIZE would not accept an allocation of more than 32GB. This limitation was due to a bug that truncated the internal value for the InnoDB buffer pool size to 32 bits on 64-bit Windows builds. (Bug #16391722, Bug #68470)
- **InnoDB:** An ALTER TABLE operation that performed a table copy failed because a temporary tablespace with the same name already existed. This fix makes temporary tables and tablespace names more unique by adding the current log sequence number (LSN) to the end of the previous table or file name. For example, table name "test/#sql-ib21" becomes "test/#sql-ib21-1701208," where 1701208 is the current LSN. Both the LSN and the table ID are needed to ensure that the name is unique because it is theoretically possible for multiple threads to have the same LSN. Including the table ID allows the temporary name to be associated with the table. (Bug #16403420)
- **InnoDB:** Crash recovery would fail with a !recv_no_log_write assertion when reading a page. (Bug #16405422)
- **InnoDB:** Restarting InnoDB in read-only mode and running a workload would occasionally return a global_segment < os_aio_n_segments assertion. (Bug #16362046)
- **InnoDB:** When the InnoDB shutdown mode (innodb_fast_shutdown) is set to 2 and the master thread enters the flush loop, the thread would not be able to exit under some circumstances. This could lead to a shutdown hang. (Bug #16411457)
- **InnoDB:** Creating a foreign key constraint using the ALTER TABLE INPLACE algorithm requires foreign_key_checks to be set to 0 (SET foreign_key_checks = 0;). As a result, an appropriate duplicate ID check would not be performed. (Bug #16413976)
- **InnoDB:** This fix removes dated and incomplete code that is protected by the UNIV_LOG_ARCHIVE macro. (Bug #16296837)
- **InnoDB:** RENAME TABLE would result in a hang due to a MySQL mutex acquisition deadlock. (Bug #16305265)
- **InnoDB:** DROP DATABASE failed if the database contained an InnoDB table that had a data file in an external data directory. The external data file had an "InnoDB Symbolic Link" file type (.isl) that was not recognized by MySQL. This fix adds .isl as a known InnoDB file type. (Bug #16338667)
- **InnoDB:** When tables are linked by foreign key constraints, loading one table would open other linked tables recursively. When numerous tables are linked by foreign key constraints, this would sometimes lead to a thread stack overflow causing the server to exit. Tables linked by foreign key constraints are now loaded iteratively. Cascade operations, which were also performed in a recursive manner, are now performed iteratively using an explicit stack. (Bug #16244691, Bug #65384)
- **InnoDB:** Under testing, a FLUSH TABLE operation resulted in a timeout due to a missing acknowledgment that the purge thread had stopped. (Bug #16277387)
- **InnoDB:** For a compressed table, the "page reorganize" function would ignore the innodb_log_compressed_pages option and always log the entire compressed page, which increased the size of the redo log. The "page reorganize" function now adheres to the

innodb_log_compressed_pages option and does not log compressed page images to the redo log when innodb_log_compressed_pages is set to “OFF”. (Bug #16267120)

- **InnoDB:** After disabling foreign key checks with SET foreign_key_checks=0 and performing a DROP INDEX, the table was no longer accessible after restarting the server. This fix allows the table with missing foreign key indexes to be accessed when SET foreign_key_checks=0. When the table is accessible, the user must recreate the missing indexes to fulfill the foreign key constraints. (Bug #16208542, Bug #68148)

- **InnoDB:** When a transaction is in READ COMMITTED isolation level, gap locks are still taken in the secondary index when a row is inserted. This occurs when the secondary index is scanned for duplicates. The function row_ins_scan_sec_index_for_duplicate() always calls the function row_ins_set_shared_rec_lock() with LOCK_ORDINARY irrespective of the transaction isolation level. This fix modifies the row_ins_scan_sec_index_for_duplicate() function to call row_ins_set_shared_rec_lock() with LOCK_ORDINARY or LOCK_REC_NOT_GAP, based on the transaction isolation level. (Bug #16133801, Bug #68021)

- **InnoDB:** Starting mysqld with --innodb_log_buffer_size=50GB failed to allocate memory and returned NULL. For non-debug builds there was no check in place and a segmentation fault occurred. This fix adds a log message stating that memory failed to be allocated, and adds an assertion. (Bug #16069598, Bug #68025)

- **InnoDB:** When UNIV_DEBUG is enabled in debug builds, buf_validate() is often called which sometimes results in false alarms in tests on semaphore wait timeout. This fix increases counter values to reduce false alarms. (Bug #16068056)

- **InnoDB:** While processing read-write workloads, InnoDB would scan more pages than are required for flushing, unnecessarily consuming CPU resource. (Bug #16037180)

- **InnoDB:** The explain_filename function, which provides information about a partition by parsing the file name, would return an error when attempting to parse a file name with no partition information. (Bug #16051728)

- **InnoDB:** Stopping the server, removing a database table (d1.t1) .frm file from the data directory, restarting the server, and dropping the database (d1), would cause an assertion. (Bug #16043216)

- **InnoDB:** While printing a UTF-8 table name, InnoDB would truncate the table name, resulting in an incomplete buffer and subsequent Valgrind error. This bug fix also addresses an incorrect debugging error message. (Bug #16066351)

- **InnoDB:** TRUNCATE TABLE would fail to handle the return value from btr_create when btr_create is invoked by TRUNCATE TABLE for creation of a new index. (Bug #16026889)

- **InnoDB:** Persistent stats would be disabled unnecessarily when running in read-only mode. When running in read-only mode, fetching stats from disk does not involve any modification of on-disk data except for when ANALYZE TABLE is run. This fix enables persistent stats for read-only mode. (Bug #16083211)

- **InnoDB:** An overflow would occur for innodb_row_lock_time_max and innodb_row_lock_current_waits. This fix modifies code logic in storage/innobase/srv/srv0srv.c. (Bug #16005310)

- **InnoDB:** Attempting to create a table while in innodb_read_only mode would result in the following error: ERROR 1015 (HY000): Can't lock file (errno: 165 - Table is read only). (Bug #15963619)

- **InnoDB:** An active FLUSH TABLE FOR EXPORT thread would cause a hang during shutdown. The fix ensures that trx_is_interrupted() is checked during ibuf_merge. (Bug #15953255)

- **InnoDB:** innochecksum would return an error when run on compressed tables. (Bug #14612872, Bug #66779)

- **InnoDB:** A multi-row INSERT ... ON DUPLICATE KEY UPDATE insert failure, caused by a duplicate key error, would result in duplicate auto-increment values. (Bug #14483484, Bug #66301)

- **InnoDB:** A mismatch between .ibd files and the InnoDB data dictionary could occur if TRUNCATE TABLE is interrupted by a crash. The mismatch would be encountered after recovery. To avoid this problem, truncate table information is written to a truncate log file that resides temporarily in the log directory. The truncate log file has the following naming convention: ib_*space_id*_trunc.log. If the truncate TRUNCATE operation is successful, the truncate log file is removed. If the TRUNCATE operation is interrupted by a crash, information is read from the truncate log file during recovery, the log records are applied, and the truncate log file is removed. (Bug #14174004, Bug #13997329, Bug #17227149, Bug #17238361)

- **InnoDB:** The documentation incorrectly stated that START TRANSACTION WITH CONSISTENT SNAPSHOT provides a consistent snapshot only if the current isolation level is REPEATABLE READ or SERIALIZABLE. START TRANSACTION WITH CONSISTENT SNAPSHOT only works with REPEATABLE READ. All other isolation levels are ignored. The documentation has been revised and a warning is now generated whenever the WITH CONSISTENT SNAPSHOT clause is ignored. (Bug #14017206, Bug #65146)

- **InnoDB:** The srv_master_thread background thread, which monitors server activity and performs activities such as page flushing when the server is inactive or in a shutdown state, runs on a one second delay loop. srv_master_thread would fail to check if the server is in a shutdown state before sleeping. (Bug #13417564, Bug #63276)

- **InnoDB:** In the error log, a full-text search index would be reported missing from the data dictionary during a TRUNCATE TABLE operation. After restarting mysqld, the following InnoDB error would be reported: “InnoDB: Error: trying to load index idx13 for table test/g1 but the index tree has been freed..” (Bug #12429565)

  References: See also Bug #17402002.

- **InnoDB:** Compiling InnoDB on Windows Vista 64-bit with VS2005 would result in compilation errors. (Bug #11752731, Bug #44004)

- **InnoDB:** When the value provided for innodb_buffer_pool_size on 32-bit systems is too large, an error message would incorrectly reference the internal variable, innobase_buffer_pool_size, instead of innodb_buffer_pool_size. (Bug #11759578, Bug #51901)

- **InnoDB:** In many cases InnoDB calls exit(1) when it encounters a fatal error. An exit(1) call does not produce a crash dump or provide information about the process state. Additionally, on Windows, an exit(1) call does not report a crashed process in the Event Viewer. This fix replaces exit(1) calls with ut_error calls in a number of places. (Bug #56400, Bug #11763660)

- **InnoDB:** Creating a table with a comment or default textual value containing an apostrophe that is escaped with a backslash would sometimes cause the InnoDB storage engine to omit foreign key definitions. (Bug #61656, Bug #12762377)

- **InnoDB:** Setting foreign_key_checks=0 and running ALTER TABLE to change the character set of foreign key columns for a database with multiple tables with foreign key constraints would leave the database in an inconsistent state. Subsequent ALTER TABLE operations (using the COPY algorithm) with foreign_key_checks=1 would fail due to the detected inconsistency. Reversion of the partially executed ALTER TABLE operation would also fail, resulting in the loss of the table being altered. When running the same ALTER TABLE operation with a RENAME clause, the inconsistency would not be detected but if the `ALTER TABLE` operation failed for some other reason, reversion of the partially executed `ALTER TABLE` would fail with the same result.

The bug fix temporarily disables `foreign_key_checks` while the previous table definition is restored. (Bug #65701, Bug #14227431)

- **InnoDB:** Successive deletes in descending key order would lead to under-filled `InnoDB` index pages. When an `InnoDB` index page is under-filled, it is merged with the left or right sibling node. The check performed to determine if a sibling node is available for merging was not functioning correctly. (Bug #68501, Bug #16417635)

- **InnoDB:** The `pthread_mutex`, `commit_threads_m`, which was initialized but never used, has been removed from the code base. (Bug #60225, Bug #11829813)

- **InnoDB:** When running an `InnoDB` full-text search in boolean mode, prefixing an asterisk (`*`) to a search string (`'*string'`) would result in an error whereas for `MyISAM`, a prefixed asterisk would be ignored. To ensure compatibility between `InnoDB` and `MyISAM`, `InnoDB` now handles a prefixed asterisk in the same way as `MyISAM`. (Bug #68948, Bug #16660607)

- **InnoDB:** The `row_check_index_for_mysql` method, which checks for NULL fields during an index scan or `CHECK TABLE` operation, would iterate unnecessarily. Thanks to Po-Chun Chang for the patch to correct this issue. (Bug #69377, Bug #16896647)

- **Partitioning:** Creating a table `t1` using `CREATE TABLE ... PARTITION BY LIST ... PARTITION ... VALUES IN (NULL)`, then attempting to execute `CREATE TABLE ... LIKE t1` caused the server to fail. (Bug #16860588)

- **Partitioning:** When upgrading to MySQL 5.5.31 or higher, a message is written into the output of `mysql_upgrade` when encountering a partitioned table for which the `ALGORITHM` option is required to maintain binary compatibility with the original; the message includes the `ALTER TABLE` statement required to make the change. For such a table having a sufficiently large number of partitions, the message was truncated with an error before the complete `ALTER TABLE` statement could be written. (Bug #16589511)

- **Partitioning:** When a range specified in the `WHERE` condition of a query against a table partitioned by `RANGE` entirely within that of one of the partitions, the next partition was also checked for rows although it should have been pruned away.

  Suppose we have a range-partitioned table `t` created using the following SQL statement:

  ```
  CREATE TABLE t  (
      id INT AUTO_INCREMENT,
      dt DATETIME,
      PRIMARY KEY (dt,id),
      UNIQUE KEY (id,dt)
  )
      PARTITION BY RANGE  COLUMNS(dt) (
          PARTITION p0 VALUES LESS THAN ('2013-01-01'),
          PARTITION p1 VALUES LESS THAN ('2013-01-15'),
          PARTITION p2 VALUES LESS THAN ('2013-02-01'),
          PARTITION p3 VALUES LESS THAN ('2013-02-15'),
          PARTITION pmax VALUES LESS THAN (MAXVALUE)
  );
  ```

  An example of a query that exhibited this issue when run against `t` is shown here:

  ```
  SELECT COUNT(*) FROM t
      WHERE dt >= '2013-02-01' AND dt < '2013-02-15';
  ```

In this case, partition pmax was checked, even though the range given in the WHERE clause lay entirely within partition p3. (Bug #16447483)

- **Partitioning:** When dropping a partitioned table, the table's .par file was deleted first, before the table definition or data. This meant that, if the server failed during the drop operation, the table could be left in an inconsistent state in which it could neither be accessed nor dropped.

  The fix for this problem makes the following changes:

  - Now, when dropping a partitioned table, the table's .par file is not removed until all table data has been deleted.
  - When executing DROP TABLE of a partitioned table, in the event that its .par file is determined to be missing, the table's .frm file is now immediately deleted, in effect forcing the drop to complete.

  (Bug #13548704, Bug #63884)

- **Replication:** The server attempted to perform an internal truncatation of the slave_worker_info table while resetting it, even though this is a DDL operation and should not be used conccurrently with DML operations. To prevent this from happening, the reset now performs sequential row deletion in place of the truncation operation. (Bug #17286858, Bug #69898)

- **Replication:** The data size for a table map event created during execution was calculated, but not when the event was created from a network packet. This could later cause problems when the data fields of such events were treated as if they had a length equal to 0 when trying to write the events to a cache, or to the binary log.

  To avoid future problems of this nature, the table map's data size is now calculated in both cases. (Bug #17164074)

- **Replication:** When the --relay-log-info-file option was used together with --slave-parallel-workers set to a value greater than 1, mysqld failed to start. (Bug #17160671)

- **Replication:** The commit error caused by the failure of binary log rotation failure generated an incident event in the binary log file and interrupted the user session with error messages which did not mention that the slave server would be stopped later when the incident event was replayed.

  Now, when encountering binlog rotation failure, a more helpful error message is instead written to the log, alerting the user to investigate in a timely manner. (Bug #17016017)

- **Replication:** The mysqlbinlog option --rewrite-db caused USE statements to be ignored, even for databases that were not referenced by the option. (Bug #16914535)

- **Replication:** The condition leading to the issue fixed in Bug #16579083 continued to raise an error even though the condition itself no longer cause the issue to occur. (Bug #16931177, Bug #69369)

  References: See also Bug #16271657, Bug #16491597, Bug #68251, Bug #68569.

- **Replication:** When rpl_semi_sync_master_timeout was set to an extremely large value, semisynchronous replication became very slow, especially when many sessions were working in parallel. It was discovered that the code to calculate this timeout was inside the wait loop itself, with the result that an increase in the value of rpl_semi_sync_master_timeout caused repeated iterations. This fix improves the method used to calculate wakeup times, and moves it outside of the wait loop, so that it is executed one time only. (Bug #16878043, Bug #69341)

- **Replication:** It was possible in CHANGE MASTER TO statements to set the MASTER_DELAY option greater than the supported maximum value ($2^{31} - 1$). In addition, the error resulting from setting

MASTER_DELAY to a value greater than 2^{32} was not handled correctly. (Bug #16820156, Bug #1690315, Bug #69249, Bug #69469)

- **Replication:** It was possible to cause a deadlock after issuing FLUSH TABLES WITH READ LOCK by issuing STOP SLAVE in a new connection to the slave, then issuing SHOW SLAVE STATUS using the original connection.

  The fix for this includes the addition of the rpl_stop_slave_timeout system variable, to control the time in seconds to wait for slave to stop after issuing STOP SLAVE before returning a warning. (Bug #16856735)

- **Replication:** Some expressions employing variables were not handled correctly by LOAD DATA. (Bug #16753869)

- **Replication:** In some circumstances, the message in the Last_Error column from the output of SHOW SLAVE STATUS referred to GTID_NEXT_LIST although this variable is not currently implemented (the name is reserved for possible future use). Now in such cases the error message no longer refers to this variable. (Bug #16742886, Bug #69096)

  References: See also Bug #16715809, Bug #69045.

- **Replication:** Point-in-time recovery could fail when trying to restore a single database from a binary log in row-based format using mysqlbinlog with the --database option. (Bug #16698172)

- **Replication:** mysqlbinlog --rewrite-db failed when the name of the destination database contained any underscore (_) characters. (Bug #16737279)

- **Replication:** Issuing a FLUSH TABLES statement on a GTID-enabled master caused replication to fail. It was found that this misbehavior was introduced by the fix for Bug #16062608, which disallowed statements that perform an implicit commit but whose changes are not logged when gtid_next is set to any value other than AUTOMATIC. The changes made in that fix have been reverted, and such statements are (again) allowed without regard to the value of this variable. (Bug #16715809, Bug #69045)

- **Replication:** On Windows platforms, issuing SHOW SLAVE STATUS while the slave I/O thread was being terminated due to an error caused the slave to fail. (Bug #16662771)

- **Replication:** A crash-on-commit error caused InnoDB to lose the previous transaction following execution of a RESET MASTER statement. This occurred because the prepare phase caused a flush to disk, while the commit phase did not perform a corresponding flush within InnoDB.

  To fix this problem, RESET MASTER now causes storage engine logs to be flushed on commit. (Bug #16666456, Bug #68932)

- **Replication:** When used with the options --dump-slave --include-master-host-port, mysqldump printed the port number within quotation marks, as if it were a string value rather than an integer. (Bug #16615117)

- **Replication:** When processing an Update_rows_log_event or Delete_rows_log_event from the binary log, the before image is hashed and stored in a hash table. Following this, the original table is scanned for the desired records; subsequent processing hashes each record fetched from the original table and performs a lookup for it in the hash table. However, columns read from the image that had originally been set to NULL could instead contain random or “garbage” data, causing the lookup (and thus replication) to fail with an error such as Could not execute Update_rows event on table.... (Bug #16621923)

  References: See also Bug #11766865. This bug was introduced by Bug #16566658.

- **Replication:** A session attachment error during group commit causes the rollback of the transaction (as intended), but the transaction in which this happened was still written to the binary log and replicated to the slave. Thus, such an error could lead to a mismatched master and slave.

  Now when this error occurs, an incident event is written in the binary log which causes replication to stop, and notifies the user that redundant events may exist in the binary log. An additional error is also now reported to the client, indicating that the ongoing transaction has been rolled back. (Bug #16579083)

- **Replication:** The error displayed by `SHOW SLAVE STATUS` when a worker thread fails to apply an event contained no event coordinate information. The GTID for the event's group was also not shown. Now in such cases, the text shown for `Last_SQL_Error` is prefixed with the (physical) master binary log coordinates, as well as the value of `gtid_next` when this has been set. (Bug #16594095)

- **Replication:** Due to time resolution issues on some systems, the time to be taken by the dump thread for a reply from the slave could be calculated to be less than zero, leading to `Semi-sync master wait for reply fail to get wait time` errors. Since this condition does not have a negative impact on replication, errors caused by these conditions have been reduced to warnings. (Bug #16579028)

- **Replication:** Linker errors occurred if the header file `log_event.h` was included in an application containing multiple source files, because the file `rpl_tblmap.cc` was included in `log_event.h`. This fix moves the inclusion of `rpl_tblmap.cc` into the source files that use `log_event.h`. (Bug #16607258)

- **Replication:** When one or more GTID log events but no previous GTIDs log events were found in the binary log, the resulting error was mishandled and led to a failure of the server. (This is an extremely rare condition that should never occur under normal circumstances, and likely indicates that the binary log file has somehow been corrupted.) Now in such cases, an appropriate error is issued, and is handled correctly. (Bug #16502579, Bug #68638)

- **Replication:** Running the server with `--log-slave-updates` together with `--replicate-wild-ignore-table` or `--replicate-ignore-table` in some cases caused updates to user variables not to be logged. (Bug #16541422)

- **Replication:** When using `mysqlbinlog` and the `mysql` client to roll forward two or more binary logs on a server having GTIDs enabled, the `gtid_next` variable was not properly reset when switching from the first to the second binary log, causing processing to halt with an error at that point. (Bug #16532543)

- **Replication:** The `mysqlbinlog` options `--include-gtids`, `--exclude-gtids`, and `--skip-gtids` did not work correctly when trying to process multiple files. (Bug #16517775)

- **Replication:** The warning issued when specifying `MASTER_USER` or `MASTER_PASSWORD` with `CHANGE MASTER TO` was unclear for a number of reasons, and has been changed to read, `Storing MySQL user name or password information in the master info repository is not secure and is therefore not recommended. Please consider using the USER and PASSWORD connection options for START SLAVE; see 'START SLAVE Syntax' in the MySQL Manual for more information.` (Bug #16460123, Bug #16461303, Bug #68602, Bug #68599)

- **Replication:** When the size of an execution event exceeded the maximum set for the buffer (`slave_pending_jobs_size_max`), row-based replication could hang with `Waiting for slave workers to free pending events`. (Bug #16439245, Bug #68462)

- **Replication:** Extra binary log rotations were performed due to concurrent attempts at rotation when the binary log became full, which were allowed to succeed. This could lead to the unnecessary creation of many small binary log files. (Bug #16443676, Bug #68575)

- **Replication:** Attempting to execute START SLAVE after importing new slave_master_info and slave_relay_log_info tables failed with an empty error message. Now an appropriate error and message are issued in such cases. (Bug #16475866, Bug #68605)

- **Replication:** Restarting the server after the slave_relay_log_info table had been emptied caused mysqld to fail while trying to return an error. (Bug #16460978, Bug #68604)

- **Replication:** Following disconnection from the master, the slave could under certain conditions report erroneously on reconnection that it had received a packet that was larger than slave_max_allowed_packet, causing replication to fail. (Bug #16438800, Bug #68490)

- **Replication:** A slave using row-based replication was unable to read the rows containing columns of type MYSQL_TYPE_DECIMAL properly (old-style decimal, used prior to MySQL 5.0.3). Now the slave throws an error if it receives this type of data. You can convert the old-style DECIMAL format to the binary format used in current MySQL releases with ALTER TABLE; see Upgrading from MySQL 4.1 to 5.0, for more information. (Bug #16416302)

- **Replication:** An SQL thread error during MTS slave recovery caused the slave to fail. (Bug #16407467, Bug #68506)

- **Replication:** DROP TEMP TABLE IF EXISTS statements could lead to failures in applying the binary log during point-in-time recovery operations. This is due to the fact that, when using row-based replication, the server appends IF EXISTS to any DROP TEMPORARY TABLE statements written to the binary log, and that the slave SQL thread does not check * wildcard filter rules for DROP TEMPORARY TABLE IF EXISTS. If --log-slave-updates was also enabled on the slave, such a statement was preceded by a USE statement. If the database referred by the USE statement did not exist, the statement failed, and stopped replication.

  Now, when writing DROP TEMPORARY TABLE IF EXISTS into the binary log, no USE statement is written, and the table name in the DROP TEMPORARY TABLE statement is a fully qualified table name. (Bug #16290902)

- **Replication:** When using the options --read-from-remote-server --stop-never --base64-output=decode-rows --verbose, mysqlbinlog failed to reset the counter used to store the current position within the file when the binary log on the server was rotated. (Bug #16316123, Bug #68347)

- **Replication:** When using mysqldump to back up a database created with MySQL 5.6.4 or an earlier version, setting --set-gtid-purged=AUTO caused the backup to fail, because pre-5.6.5 versions of MySQL did not support GTIDs, and it could not be determined if GTIDs were enabled for the database. This fix makes sure mysqldump does not attempt to output a SET @@global.gtid_purged statement when backing up any pre-5.6.5 databases. (Bug #16303363, Bug #68314)

- **Replication:** After a transaction was skipped due to its GTID already having been logged, all remaining executed transactions were incorrectly skipped until gtid_next was pointed to a different GTID.

  To avoid this incorrect behavior, all transactions—even those that have been skipped—are marked as undefined when they are commited or rolled back, so that an error is thrown whenever a second transaction is executed following the same SET @@session.gtid_next statement. (Bug #16223835)

- **Replication:** Deadlocks could sometimes occur on group commits with a high number of concurrent updates, as well as when one client held a lock from a commit while another client imposed a lock while rotating the binary log. (Bug #16271657, Bug #16491597, Bug #68251, Bug #68569)

- **Replication:** Modifying large amounts of data within a transaction can cause the creation of temporary files. Such files are created when the size of the data modified exceeds the size of the binary log cache (max_binlog_cache_size). Previously, such files persisted until the client connection was closed, which could allow them to grow until they exhausted all available disk space in tmpdir. To prevent this from occurring, the size of a temporary file created in this way in a given transaction is now reset to 0 when the transaction is committed or rolled back. (Bug #15909788, Bug #18021493, Bug #66237)

- **Replication:** When semisynchronous replication was enabled, the automatic dropping on the master of an event created using `ON COMPLETION NOT PRESERVE` caused the master to fail. (Bug #15948818, Bug #67276)
- **Replication:** When the master had more than one table with an auto-increment column, *and* the slave ignored at least one of these tables due to `--replicate-ignore-table` rules, *but* at least one them was replicated, even so—the replicated table or tables having at least one trigger updating one or more tables existing only on the slave—updates to any of the auto-increment tables on the master caused replication to fail. (Bug #15850951, Bug #67504)
- **Replication:** Setting a `SET` column to `NULL` inside a stored procedure caused replication to fail. (Bug #14593883, Bug #66637)
- **Replication:** The binary log contents got corrupted sometimes, because the function `MYSQL_BIN_LOG::write_cache` always thought it had reached the end-of-cache when the function `my_b_fill()` reported a '0,' while that could also mean an error had occurred. This fix makes sure that whenever `my_b_fill()` returns a '0,' an error check is performed on `info->error`. (Bug #14324766, Bug #60173)
- **Replication:** The internal function `MYSQL_BIN_LOG::open_binlog()` contained an unneeded variable, which has been removed. (Bug #14134590, Bug #60188)
- **Replication:** `PURGE BINARY LOGS` by design does not remove binary log files that are in use or active, but did not provide any notice when this occurred. Now, when log files are not removed under such conditions, a warning is issued; this warning includes information about the file or files were not removed when the statement was issued. (Bug #13727933, Bug #63138)
- **Replication:** It was possible for the multi-threaded slave coordinator to leak memory when the slave was stopped while waiting for the next successful job to be added to the worker queue. (Bug #13635612)
- **Replication:** When replicating to a `BLACKHOLE` table using the binary logging format, updates and deletes cannot be applied and so are skipped. Now a warning is generated for this whenever it occurs.

  Note

  `binlog_format=STATEMENT` is recommended when replicating to tables that use the `BLACKHOLE` storage engine.

  (Bug #13004581)
- **Replication:** Temporary files created by `LOAD DATA INFILE` were not removed if the statement failed. (Bug #11763934, Bug #56708)
- **Replication:** After the client thread on a slave performed a `FLUSH TABLES WITH READ LOCK` and was followed by some updates on the master, the slave hung when executing `SHOW SLAVE STATUS`. (Bug #68460, Bug #16387720)
- **Microsoft Windows:** On Microsoft Windows, passing in `--local-service` to `mysqld.exe` when also passing in a service name could cause a crash at startup. (Bug #16999777, Bug #69549)
- The contents of SQL condition items such as `TABLE_NAME`, `CONSTRAINT_NAME`, an so forth were lost if resignaled by a stored routine condition handler. (Bug #17280703)
- `SELECT * from performance_schema.events_statements_current` could raise an assertion due to a race condition under load. (Bug #17164720)

- AES_ENCRYPT() and AES_DECRYPT() failed to work correctly when MySQL was built with an AES_KEY_LENGTH value of 192 or 256. (Bug #17170207)
- InnoDB full-text searches failed in databases whose names began with a digit. (Bug #17161372)
- A successful connection failed to reset the per-IP address counter used to count successive connection failures. This could possibly cause a host to be blocked, when the max_connect_errors limit was reached. (Bug #17156507)
- Under load, truncating the accounts Performance Schema table could cause a server exit. (Bug #17084615)
- With the thread pool plugin enabled and SSL in use, an error in one connection might affect other connections, causing them to experience a lost connection. (Bug #17087862)
- Indexed lookups on POINT columns was slower for InnoDB tables in MySQL 5.7 compared to 5.6. (Bug #17057168)
- The Performance Schema was built for embedded server builds. This no longer occurs. (Bug #17041705)
- my_pthread.h unconditionally included pfs_thread_provider.h, a noninstalled header file, resulting in compilation failure when compiling MySQL applications against the installed header files. (Bug #17061480)
- For debug builds, improper use of SAFE_MUTEX within dbug.c caused different code areas to have different ideas about size and contents of a mutex. This could result in out-of-bounds memory writes. (Bug #16945343)
- Initialization of keycache_* variables (see Multiple Key Caches) during server startup could write to incorrect memory. (Bug #16945503)
- Reads from message buffers for closed connections could occur. (Bug #17003702)
- At server startup, it was possible to set the validate_password_length system variable to a value smaller than permitted by the values of other password-length variables related to it. (Bug #16957721)
- The server could exit while using a cursor to fetch rows from a UNION query. (Bug #16983143)
- The range optimizer incorrectly assumed that any geometry function on a spatial index returned rows in ROWID order, which could result in incorrect query results. (Bug #16960800)
- A race condition in the thread pool plugin could cause status variables such as Aborted_connects not to be incremented and permitting concurrent kills to happen for the same thread ID. (Bug #16959022)
- mysql_secure_installation did not properly clean up the mysql.proxies_privs table for removed accounts. (Bug #16959850)
- The server did excessive locking on the LOCK_active_mi and active_mi->rli->data_lock mutexes for any SHOW STATUS LIKE 'pattern' statement, even when the pattern did not match status variables that use those mutexes (Slave_heartbeat_period, Slave_last_heartbeat, Slave_received_heartbeats, Slave_retried_transactions, Slave_running). Now attempts to show those variables do not lock those mutexes. This might result is slightly stale data, but better performance. (Bug #16904035)
- Full-text phrase search in InnoDB tables could read incorrect memory. (Bug #16885178)
- It was not possible to keep several major versions of MySQL in the same yum repository. (Bug #16878042)

- The Performance Schema could spawn a thread using incorrect instrumentation information. (Bug #16939689)
- The Batched Key Access method could return incorrect results on big-endian machines if a table buffered in the BKA join cache used a storage engine such as InnoDB or MyISAM with little-endian storage format, and then the next table used a storage engine such as NDB with native-endian storage format. (Bug #16853897)
- The error string for ER_COL_COUNT_DOESNT_MATCH_PLEASE_UPDATE string contained a hardcoded database name ('mysql.%s'), which is incorrect when the error referred to a table in a different database. (Bug #16813605)
- Excessive memory consumption was observed for multiple execution of a stored procedure under these circumstances: 1) The stored procedure had an SQL statement that failed during validation. 2) The stored procedure had an SQL statement that required repreparation. (Bug #16857395)
- HAVE_REPLICATION now is set from CMake rather than in my_global.h so that it is not dependent on my_global.h having been included. (Bug #16768511)
- CMake now assumes the existence of standard C header files such as stdlib.h and stdarg.h. (Bug #16748528)
- Some errors in MySQL 5.7 had different numbers than in MySQL 5.6. (Bug #16780120)
- Removing a server RPM package did not shut down the existing server if it was running. (Bug #16798868)
- INSERT ... ON DUPLICATE KEY UPDATE could cause a server exit if a column with no default value was set to DEFAULT. (Bug #16756402)

  References: This bug is a regression of Bug #14789787.
- An assertion could be raised when the optimizer considered pushing down an index condition containing an updatable user variable and did not contain fields from the index. (Bug #16804581)
- The function fill_locks_row(), which is responsible for providing data for the INFORMATION_SCHEMA.INNODB_LOCKS table, would try to look up the B-tree page in the buffer pool for INFIMUM and SUPREMUM records, both of which have a predefined heap_no. This generated unnecessary buffer pool contention and caused information to be omitted when a page was not available in the buffer pool. This fix removes the buffer pool lookup for PAGE_HEAP_NO_INFIMUM (heap_no=0) and PAGE_HEAP_NO_SUPREMUM (heap_no=1) from fill_locks_row(). (Bug #16684523)
- The code base was modified to account for new warning checks introduced by gcc 4.8. (Bug #16729109)
- Compiling failed with -DMY_ATOMIC_MODE_RWLOCKS=1 or on platforms on which MySQL did not support lockless atomic operations (such as ARM). (Bug #16736461)
- In a prepared statement or stored routine, if the HAVING clause of a subquery referenced some column of the GROUP BY of the parent query, the server could exit. (Bug #16739050)
- Password rewriting in the general query log now also applies to prepared statements. (Bug #16732621)
- The deprecated thread_concurrency system variable has been removed. (Bug #16661195)
- Overhead for setting PROCESSLIST_STATE values in the Performance Schema THREADS table has been reduced. (Bug #16633515)

- The read-only open_files_limit system variable did not show the maximum number of open files the mysqld process could have, but instead the number that was requested after adjusting the --open-files-limit command-line option. (Bug #16657588)
- Kill handling in the thread pool plugin was subject to timeout problems and Valgrind warnings. (Bug #16633880)
- Within a stored procedure, repeated execution of a prepared CREATE TABLE statement for a table with partitions could cause a server exit. (Bug #16614004)
- Some problems compiling on Solaris in 64-bit mode with gcc and g++ were corrected. (Bug #16555106)
- The Windows authentication plugin failed to free a context buffer for each connection. (Bug #16591288)
- The DBUG_PRINT() macro unnecessarily evaluated arguments when debugging was not enabled. (Bug #16556597)
- Some rows for a session could be missing sporadically from the Performance Schema session_connect_attrs table while the session was executing a workload. (Bug #16576980)
- The server could make the wrong decision about whether an account password was expired. (Bug #16604641)
- Enabling the query cache could cause repeatable-read transactions to return incorrect results. (Bug #16497925)
- When index condition pushdown was used on a descending range scan and the first range interval did not contain any qualifying records, the result of the range scan could be empty even if other range intervals contained qualifying records. (Bug #16483273)
- The server could attempt a filesort operation for a zero-size sort length, causing it to exit. (Bug #16503160)
- SHOW WARNINGS and SHOW ERRORS did not properly reset the warning count. (Bug #16522662)
- my_load_defaults() was modified to accommodate some problems under compilation with gcc 4.7.2 that could cause a client crash during option processing. (Bug #16497125)
- Opening a cursor on a SELECT within a stored procedure could cause a segmentation fault. (Bug #16499751)

  References: This bug is a regression of Bug #14740889.
- Geometry methods that worked with WKB data performed insufficient input data validation, which could cause Valgrind errors or a server exit. (Bug #16510712, Bug #12772601)
- Missing variable initialization led to incorrectly returning an error from st_select_lex_unit::explain and led to a failed assertion. (Bug #16484966)
- Clients could determined based on connection error message content whether an account existed. (Bug #16513435)
- CREATE TABLE or ALTER TABLE can fail if the statement specified unsupported options or something was missing. Previously, such errors were returned as ER_ILLEGAL_HA. Now they are returned as the new ER_MISSING_HA_CREATE_OPTION error. (Bug #16498740)
- Failure to handle a full-text search wildcard properly could cause the server to exit. (Bug #16446108)
- Some INFORMATION_SCHEMA queries that used ORDER BY did not use a filesort optimization as they did in MySQL 5.5. (Bug #16423536)

- Performance Schema parameter autosizing at startup did not take into account later autosizing changes to other startup parameters on which the Performance Schema parameters depended. (Bug #16430532)
- The WKB reader for spatial operations could fail and cause a server exit. (Bug #16451878)
- For queries using `ref` access on `CHAR` and `VARCHAR` data types, the `ref` access condition could be evaluated again as part of the query condition or pushed down as an index condition to the storage engine. (Bug #16437630)
- A full-text search syntax error failed to print to standard output. (Bug #16429688, Bug #16765397)
- Optimizer heuristics inappropriately preferred range access over `ref` access in cases when the `ref` access referred to a column of a table earlier in the join seqence. (Bug #16437940)
- The `ER_OUTOFMEMORY` error was used in some places without the proper parameter being passed, resulting in incorrect diagnostic output. (Bug #16449659)
- Unlike `MyISAM`, `InnoDB` does not support boolean full-text searches on nonindexed columns, but this restriction was not enforced, resulting in queries that returned incorrect results. (Bug #16434374)
- If a table has been marked as containing only `NULL` values for all columns if it is a `NULL`-complemented row of an outer join, then rollup on the column which cannot be nullable results in a server exit. (Bug #16436014)
- If the optimizer was using a loose index scan, the server could exit while attempting to create a temporary table. (Bug #16436567)
- Incorrect results or a server exit could be caused by a reference to an aggregated expression inside a nested subquery, where the aggregated expression was evaluated in a query block more than two levels outer to the reference. (Bug #16436383)
- In debug builds, failure in the range optimizer for an `ER_LOCK_DEADLOCK` or `ER_LOCK_WAIT_TIMEOUT` error could go undetected and cause the `filesort` code to raise an assertion. In release builds, this problem manifested as clients receiving an `ER_FILSORT_ABORT` rather than the correct error code. (Bug #16366881)
- If a lock timeout resulted from an `UPDATE` with a nested `SELECT` being unable to access rows being accessed by another thread, the error could go unchecked and cause an assertion to be raised later. (Bug #16367039)
- In debug builds, failure in the range optimizer for an `ER_LOCK_DEADLOCK` or `ER_LOCK_WAIT_TIMEOUT` error could go undetected and cause an assertion to be raised when a response was sent to the client. In release builds, this problem manifested as clients receiving an `OK` for a statement that had failed. (Bug #16366994, Bug #16247110)
- Manually-created accounts (using `INSERT`) with a malformed password effectively had no password. (Bug #16414396)
- Several scripts in the `sql-bench` directory that were supposed to be executable did not have the executable access bit set. (Bug #16395606)
- For debug builds, `DBUG_EXPLAIN` resulted in a buffer overflow when the `debug` system variable value was more than 255 characters. (Bug #16402143)
- For debug builds, with an XA transaction in IDLE or PREPARED status, execution of a query with the query cache enabled could cause a server exit. (Bug #16388996)
- A race condition in `vio_shutdown()` could cause a server exit. (Bug #16354789)

- An assertion could be raised when creating a index on a prefix of a TINYBLOB column in an InnoDB column. (Bug #16368875)
- For debug builds, set_field_to_null() could raise an assertion for attempts to insert NULL into a NOT NULL column. (Bug #16362246)
- A server exit could occur for queries of the form SELECT (SELECT 1 FROM t1) IN (SELECT a FROM t1) when attempting to evaluate the constant left-hand argument to the IN subquery predicate. (Bug #16369522)
- An assertion could be raised if, in greedy search mode, the optimizer identified join orders but was unable to choose one. (Bug #16361170)
- Debugging flags used to set the debug system variable were ignored if they were a prefix of keywords already in the debugging list. (Bug #16415978)
- If the primary key for the mysql.proc system table was removed (an unsupported and not-recommended operation), the server exited for subsequent stored procedure invocation. Similar problems could occur for other system tables. Now an error occurs instead. (Bug #16373054)
- Oracle RPM packages were unusable by yum due to issues with the obsoletes line in the .spec file causing yum to interpret the package as obsoleting itself. (Bug #16298542)
- Re-execution of a stored procedure could cause a server exit in Item_field::fix_outer_field. (Bug #16317443)
- A GROUP_CONCAT() invocation containing subquery having an outer reference caused the server to exit. (Bug #16347343)
- For debug builds, GROUP_CONCAT(... ORDER BY) within an ORDER BY clause could cause a server exit. (Bug #16347426)
- The validate_password plugin did not always enforce appropriate constraints against assigning empty passwords. (Bug #16346443)
- For debug builds, the server could exit for queries involving a nested subquery, a subquery transformed into a semi-join and using a view. (Bug #16317076)
- No warning was generated if a duplicate index existed after dropping a column associated with a multiple-column index. (Bug #16315351)
- SELECT DISTINCT with WITH ROLLUP could result in a Duplicate entry 'NULL' for key '<auto_key>' error. (Bug #16314835)
- The server could exit in do_copy_not_null() due to an improper NULL-value check. (Bug #16316564)
- Transforming some subqueries that select temporal or BIGINT types or to a semijoin caused a server exit on the second execution of prepared statements or stored programs. (Bug #16319671)
- The range optimizer could set up incorrect ranges for queries that used XOR operations. (Bug #16272562)
- mysql_secure_installation could not connect to the server if the account used had an expired password. It invoked mysql noninteractively, resulting in that program failing to connect. Now mysql supports a --connect-expired-password option that indicates to the server that it can handle sandbox mode for expired-password accounts even if invoked noninteractively, and mysql_secure_installation invokes mysql with this option. (Bug #16248315)

- If loose index scan was used on a query that used MIN(), a segmentation fault could occur. (Bug #16222245)
- The usual failed-login attempt accounting was not applied to failed COM_CHANGE_USER commands. (Bug #16241992, Bug #17357535)
- For debug builds, an assertion could be raised if a failed LOAD DATA INFILE statement will followed by an INSERT for the same table within the same session. (Bug #16240526)
- For debug builds, an assertion was incorrectly raised for queries executed using eq_ref access and filesort. (Bug #16164885)
- A user variable referenced during execution of a prepared statement is set to memory that is freed at the end of execution. A second execution of the statement could result in Valgrind warnings when accessing this memory. (Bug #16119355)
- Misoptimization of left expressions in prepared statements could cause a server exit. (Bug #16095534)
- A prepared statement that used GROUP_CONCAT() and an ORDER BY clause that named multiple columns could cause the server to exit. (Bug #16075310)
- The server could exit the second time a stored routine was invoked that performed an UPDATE or DELETE using an invalid column in the join clause. (Bug #16078466)
- If my_write() encountered a disk-full condition, it could return an incorrect error value. (Bug #16078792)
- Creating a FEDERATED table without specifying a connection string caused a server exit. (Bug #16048546)
- ORDER BY MATCH ... AGAINST could cause a server exit. (Bug #16073689)
- Certain queries containing ORDER BY or SQL_CALC_FOUND_ROWS could cause a server exit for JSON-format EXPLAIN statements. (Bug #16077396, Bug #16078113)
- Client programs from MySQL 5.6.4 and up could confuse older servers during the connection process by using newer protocol features not understood by older servers. (Bug #15965409)
- When a partition is missing, code in ha_innodb.cc would retry 10 times and sleep for a microsecond each time while holding LOCK_open. The retry logic for partitioned tables was introduced as a fix for Bug#33349 but did not include a test case to validate it. This fix removes the retry logic for partitioned tables. If the problem reported in Bug#33349 reappears, a different solution will be explored. (Bug #15973904)
- The mysql.server script exited with an error if the status command was executed with multiple servers running. (Bug #15852074)
- Use of the VALUES() function in the VALUES() clause of an INSERT statement could result in Valgrind warnings or an unstable server, possibly leading to a server exit. (Bug #14789787)
- In some cases, REVOKE could fail to revoke the GRANT OPTION privilege. (Bug #14799187)
- The Debug Sync facility could lose a signal, leading to a spurious ER_DEBUG_SYNC_TIMEOUT error. (Bug #14765080, Bug #18221750)
- The mysql client allocated but did not free a string after reading each line in interactive mode, resulting in a memory leak. (Bug #14685362)
- The optimizer trace could print ranges for key parts that were not usable for range access. (Bug #14615536)

- Killing a connection while it was in the process of disconnecting could lead to an assertion being raised, Valgrind warnings, and general unstability. (Bug #14560522)
- When running a query on `INFORMATION_SCHEMA.INNODB_BUFFER_PAGE` that requested `table_name` and `index_name` values, query results would include index pages without `table_name` or `index_name` values. (Bug #14529666)
- Several `COM_xxx` commands in the client-server protocol did not have length checks for incoming network packets, which could result in various problems for malformed input. (Bug #14525642)
- Passwords in statements were not obfuscated before being written to the audit log. (Bug #14536456)
- If used to process a prepared `CALL` statement for a stored procedure with `OUT` or `INOUT` parameters, `mysql_stmt_store_result()` did not properly set the flags required to retrieve all the result sets. (Bug #14492429, Bug #17849978)
- `INSERT ... ON DUPLICATE KEY UPDATE` on a view could cause a server exit. (Bug #14261010)
- With the thread pool plugin in use, normal connection termination caused the `Aborted_clients` status variable to be incremented. (Bug #14081240)
- A build failure occurred if `HAVE_CRYPT` was 0. (Bug #14036425)
- Grouping by an outer `BLOB` column in a subquery caused a server exit. (Bug #13966809, Bug #14700180)
- On Windows, command-line options of the form `--opt_name="opt_value"` worked but `--opt_name='opt_value'` did not.

  On all platforms, for Performance Schema options of the form `--performance_schema_instrument="instrument=value"`, invalid instrument names now are rejected. (Bug #13955232)
- The server could exit after failing to handle an out-of-memory condition in `open_normal_and_derived_tables()`. (Bug #13553905)
- The server could exit due to improper handling of the error from an invalid comparison. (Bug #13009341)
- MySQL Installer, if run in custom install or change mode, offered installation options that had no effect. (Bug #12928601)
- The thread pool plugin produced an error message containing an incorrect maximum `thread_pool_prio_kickup_timer` value. (Bug #12817590)
- Metadata returned for a prepared `SELECT` statement that had outer joins could indicate that columns containing `NULL` values were `NOT NULL`. (Bug #12818811)
- For debug builds, the server could exit as a result of a series of statements that used a user variable such that its character set/collation changed from statement to statement. (Bug #12368577)
- Incorrect results could be returned from queries that used several `aggr_func(DISTINCT)` functions (where `aggr_func()` is an aggregate function such as `COUNT()`) when these referred to different columns of the same composite key. (Bug #12328597)
- The `CMake` check for `unsigned time_t` failed on all platforms. (Bug #11766815)
- An identifier containing special characters could become too long after conversion of such characters to encoded format, resulting in SQL errors or failure to find files. (Bug #11766880)

- `mysql_convert_table_format` ignored `--engine` or `-e` as a synonym for the `--type` option. (Bug #11756950)
- `mysqladmin debug` causes the server to write debug information to the error log. On systems that supported `mallinfo()`, the memory-status part of this output was incorrect in 64-bit environments when `mysqld` consumed more than 4GB memory.

  Now the server uses `malloc_info()` to obtain memory-status information. `malloc_info()` does not report the memory that the `glibc` `malloc()` implementation internally allocates using `mmap()`. However, it does provide the memory usage information in all the memory arenas.

  This bug fix also involves a change of output format. The server now writes memory information in XML format rather than as plain text. Example:

  ```
  Memory status:
  <malloc version="1">
  <heap nr="0">
  <sizes>
  <size from="33" to="33" total="1056" count="32"/>
  <size from="65" to="65" total="65" count="1"/>
  <size from="113" to="113" total="226" count="2"/>
  <size from="129" to="129" total="2451" count="19"/>
  <size from="145" to="145" total="290" count="2"/>
  <size from="161" to="161" total="1288" count="8"/>
  <size from="209" to="209" total="418" count="2"/>
  </sizes>
  <total type="fast" count="0" size="0"/>
  <total type="rest" count="66" size="5794"/>
  <system type="current" size="10833920"/>
  <system type="max" size="10833920"/>
  <aspace type="total" size="10833920"/>
  <aspace type="mprotect" size="10833920"/>
  </heap>
  <total type="fast" count="0" size="0"/>
  <total type="rest" count="66" size="5794"/>
  <system type="current" size="10833920"/>
  <system type="max" size="10833920"/>
  <aspace type="total" size="10833920"/>
  <aspace type="mprotect" size="10833920"/>
  </malloc>
  ```

  (Bug #11746658)
- When an `ALTER TABLE` operation was performed with an invalid foreign key constraint, the error reported was `ER_CANT_CREATE_TABLE` rather than `ER_CANNOT_ADD_FOREIGN`. (Bug #64617, Bug #13840553)
- If an account had a nonzero `MAX_USER_CONNECTIONS` value, that value was not always respected. (Bug #65104, Bug #14003080)
- On 64-bit Mac OS X systems, `CMake` used `x86` rather than `x86_64` when determining the machine type. (Bug #58462, Bug #11765489)
- For failure to create a new thread for the event scheduler, event execution, or new connection, no message was written to the error log. This could lead to the impression that the event scheduler was running normally when it was not. (Bug #67191, Bug #14749800, Bug #16865959)
- Compilation on Solaris using `gcc` produced incorrect builds for 32-bit systems. (Bug #68675)
- The `mysql` client incorrectly used `latin1` for certain comparisons even if started with a multi-byte default character set, resulting in a client crash. (Bug #68107, Bug #16182919)

- In option files, the server could misinterpret option settings if the value was given after the option name with no = sign in between. (Bug #67740, Bug #15930031)

- The url columns in the mysql datatbase help tables were too short to hold some of the URLs in the help content. These columns are now created as type TEXT to accommodate longer URLs. (Bug #61520, Bug #12671635)

- mysqld --help and mysqld --verbose --help performed unnecessary logging. (Bug #68578, Bug #16442113)

- Performance of prepared DML statements containing ? parameter substitution markers was improved under row-based logging format: Since the binary log in this case need not include the statement text, and since the statement will not be forced to statement-based logging as some DDL statements might be, there is no need to substitute ? markers to produce a statement suitable for logging. (Bug #67676, Bug #16038776)

- ELT(LEAST(...),..) could return a non-NULL value even if LEAST() returned NULL. (Bug #67578, Bug #16171537)

- InnoDB does not support full-text parser plugins, but failed to report an error if they were specified. Now an ER_INNODB_NO_FT_USES_PARSER error is returned. (Bug #62004, Bug #12843070)

- If Loose Index Scan was used to evaluate a query that compared an integer column to an integer specified as a quoted string (for example, *col_name* = '1'), the query could return incorrect results. (Bug #68473, Bug #16394084)

- IF() function evaluations could produce different results when executed in a prepared versus nonprepared statement. (Bug #45370, Bug #11753852)

- If a function such as AES_DECRYPT() that requires SSL support failed, the error could affect later calls to functions that require SSL support. (Bug #68340, Bug #16315767)

- In a MySQL server newer than MySQL 5.5 using a nonupgraded mysql.user table (for which mysql_upgrade had not been run), statements to set passwords caused a server exit due to a faulty check for the password_expired column. (Bug #68385, Bug #16339767)

- It is now possible to suppress installation of the mysql-test directory after compiling MySQL from source by invoking CMake with the INSTALL_MYSQLTESTDIR option explicitly set to empty:

  ```
  cmake . -DINSTALL_MYSQLTESTDIR=
  ```

  Previously, attempts to do this resulted in an error. (Bug #58615, Bug #11765629)

- When only counting events but not timing them, Performance Schema would report MIN_TIMER_WAIT values as a large number instead of 0. (Bug #68768, Bug #16552425)

- The Turbo Boyer-Moore algorithm used for LIKE pattern matches failed to handle some patterns. The server now uses the original Boyer-Moore algorithm. (Bug #59973, Bug #11766777)

- If mysqld crashed during a shutdown initiated by /etc/init.d/mysql stop, mysqld_safe restarted mysqld when it should not have. (Bug #34084, Bug #13864548)

- Using range access with an index prefix could produce incorrect results. (Bug #68750, Bug #16540042)

- For debug builds, metadata locking for CREATE TABLE ... SELECT could raise an assertion. (Bug #68695, Bug #16503173)

- Long table or column names could cause mysqlshow to exit. (Bug #53956, Bug #11761458)

- A new CMake option, WITH_EDITLINE, is provided to indicate whether to use the bundled or system libedit/editline library. The permitted values are bundled (the default) and system.

  WITH_EDITLINE replaces WITH_LIBEDIT, which has been removed. (Bug #68558, Bug #16430208)
- When specified in an option file, the plugin-dir client option was ignored. (Bug #68800, Bug #16680313)
- Overhead for the skip_trailing_space() function was reduced. (Bug #68477, Bug #16395778)
- If an UPDATE containing a subquery caused a deadlock inside InnoDB, the deadlock was not properly handled by the SQL layer. The SQL layer then tried to unlock the row after InnoDB rolled back the transaction, raising an assertion inside InnoDB. (Bug #69127, Bug #16757869)
- Configuring MySQL with -DWITH_EXTRA_CHARSETS=none caused a build failure. (Bug #58672, Bug #11765682)
- The !includedir directive in option files did not read .cnf or .ini files that included a dot in the file name preceding the extension. (Bug #51609, Bug #11759306)
- Indexes on derived tables that were used during the first invocation of a stored procedure were not used in subsequent invocations. (Bug #68350, Bug #16346367)
- Boolean plugin system variables did not behave well on machines where char is unsigned; some code attempted to assign a negative value to these. (Bug #59905, Bug #11864205)
- For DELETE and UPDATE statements, EXPLAIN displayed NULL in the ref column for some cases where const is more appropriate. (Bug #68299, Bug #16296268)
- The optimizer could choose a poor execution plan for queries with ORDER BY ... LIMIT. (Bug #69013, Bug #16697792)
- FOUND_ROWS() could return an incorrect value if the preceding query used filesort. (Bug #69119, Bug #16760474)

  References: This bug is a regression of Bug #68458.
- In the absence of SQL_CALC_FOUND_ROWS in the preceding query, FOUND_ROWS() should return the number of rows in the result set, but this did not always happen if the query contained ORDER BY. (Bug #69271, Bug #16827872)
- sql-common/client_plugin.c contained a nonportable use of a va_list parameter. (Bug #62769, Bug #13252623)
- mysqldump assumed the existence of the general_log and slow_log tables in the mysql database. It failed if invoked to dump tables from an older server where these tables do not exist. (Bug #65670, Bug #14236170)
- Configuring with cmake -DWITHOUT_SERVER to build clients without the server failed for builds outside of the source tree. (Bug #66000, Bug #14367046)
- Full-text search on InnoDB tables failed on searches for words containing apostrophes. (Bug #69216, Bug #16801781)
- Full-text search on InnoDB tables failed on searches for literal phrases combined with + or - operators. (Bug #68720, Bug #16516193)
- Optimizations that used extended secondary keys (see Use of Index Extensions) worked only for InnoDB, even for storage engines with the requisite underlying capabilities. (Bug #68469, Bug #16391678)

- Successful queries served from the query cache did not clear warnings. (Bug #49634, Bug #11757567)
- With big_tables enabled, queries that used COUNT(DISTINCT) on a simple join with a constant equality condition on a non-duplicate key returned incorrect results. (Bug #52582, Bug #11760197)
- If ALTER TABLE was used to set the default value for a TIMESTAMP or DATETIME column that had CURRENT_TIMESTAMP as its default when it was created, the new default was not shown by SHOW CREATE TABLE, and incorrect values could be inserted into the column. (Bug #45669, Bug #11754116)
- mysql_install_db incorrectly tried to create the mysql.innodb_table_stats and mysql.innodb_index_stats tables if InnoDB was not available. (Bug #68438, Bug #16369955)
- If one session had any metadata lock on a table, another session attempting CREATE TABLE [IF NOT EXISTS] for the same table would hang. This occurred due to an attempt in the second session to acquire an exclusive metadata lock on the table before checking whether the table already existed. An exclusive metadata lock is not compatible with any other metadata locks, so the session hung for the lock timeout period if another session had the table locked.

  Now the server attempts to acquire a shared metadata lock on the table first to check whether it exists, then upgrade to an exclusive lock if it does not. If the table does exist, an error occurs for CREATE TABLE and a warning for CREATE TABLE IF NOT EXISTS. (Bug #63144, Bug #13418638)
- Attempts to build from a source RPM package could fail because the build process attempted to refer to a pb2user that might not exist. (Bug #64641, Bug #13865797, Bug #69339, Bug #16874980)
- A typo in cmake/dtrace.cmake prevented DTrace support from being enabled by -DENABLE_DTRACE-on. (Bug #60743, Bug #12325449)
- When an internal buffer was too small for the workload, the Performance Schema could spend a lot of time in an internal spin loop attempting to allocate a memory buffer, and fail. (Bug #69382, Bug #16945618)
- BIT(0) is not a valid data type specification but was silently converted to BIT(1). Now an ER_INVALID_FIELD_SIZE error occurs and the specification is rejected. (Bug #68419, Bug #16358989)
- Some LEFT JOIN queries with GROUP BY could return incorrect results. (Bug #68897, Bug #16620047)

  References: This bug is a regression of Bug #11760517.
- Some possible cases of memory use after being freed were fixed. Thanks to Jan Staněk for the patch. (Bug #68918, Bug #16725945)
- mysqldump wrote SET statements as SET OPTION, which failed when reloaded because the deprecated OPTION keyword has been removed from SET syntax. (Bug #67507, Bug #15844882)
- If the server could not find the errmsg.sys file at startup, the resulting error message did not indicate which configuration parameter to check. (Bug #67576, Bug #15880018)
- If query_cache_type was disabled at startup to prevent the query cache from being enabled at runtime, disabling query_cache_type at runtime generated a warning even though it was already disabled. (Bug #69396, Bug #16906827)
- For queries with ORDER BY ... LIMIT, the optimizer could choose a nonordering index for table access. (Bug #69410, Bug #16916596)
- When selecting a union of an empty result set (created with WHERE 1=0 or WHERE FALSE) with a derived table, incorrect filtering was applied to the derived table. (Bug #69471, Bug #16961803)

References: This bug is a regression of Bug #15848521.

- For better robustness against stack overflow, the server now accounts for the size of the guard area when making thread stack size requests. (Bug #35019, Bug #11748074)

- Two problems adding or subtracting keyword from the current debug system variable setting were corrected:

  - A debug value of 'd' means “all debug macros enabled”. The following sequence left the value in an incorrect state:

    ```
    mysql> SET debug = 'd';SELECT @@debug;
    +---------+
    | @@debug |
    +---------+
    | d       |
    +---------+

    mysql> SET debug = '+d,M1';SELECT @@debug;
    +---------+
    | @@debug |
    +---------+
    | d,M1    |
    +---------+
    ```

    The first SET statement enables all debug macros. The second SET should add the M1 macro to the current set, which should result in no change because the current set is already “all macros”. Instead, the second SET reset the current set to only the M1 macro, effectively disabling all others. The server now correctly leaves debug set to 'd'.

  - A debug value of '' means “no debug macros enabled”. The following sequence left the value in an incorrect state:

    ```
    mysql> SET debug = 'd,M1';SELECT @@debug;
    +---------+
    | @@debug |
    +---------+
    | d,M1    |
    +---------+

    mysql> SET debug = '-d,M1';SELECT @@debug;
    +---------+
    | @@debug |
    +---------+
    | d       |
    +---------+
    ```

    The first SET statement sets debug to the M1* macro. The second SET should subtract the M1 macro from the current set, leaving no debug macros enabled. Instead, the second SET reset the current set to 'd' (all macros enabled). The server now correctly sets debug to ''.

  (Bug #58630, Bug #11765644)

- Some subquery transformations were not visible in EXPLAIN output. (Bug #59852, Bug #11766685)

- Comparison of a DATETIME value and a string did not work correctly for the utf8_unicode_ci collation. (Bug #68795, Bug #16567381)

- The range optimizer used the wrong prerequisite for concluding that a table is the inner table of an outer join. This led to incorrect cost estimates and choice of the wrong index for query processing. (Bug #37333, Bug #11748775)
- Some errors could be handled by condition handlers only if they were raised by particular statements, such as INSERT, but not if they were raised by other statements, like UPDATE. An example would be the foreign-key error ER_NO_REFERENCED_ROW_2 which could be treated differently, depending on which statement raised it. (Bug #68831, Bug #16587369)
- The libmysql.dll library was missing several symbols: my_init, mysql_client_find_plugin, mysql_client_register_plugin, mysql_load_plugin, mysql_load_plugin_v, mysql_options4, and mysql_plugin_options. (Bug #69204, Bug #16797982, Bug #62394)
- If the WITH_SSL CMake option was specified with an incorrect path to the SSL installation or the path to an unsupported (too old) SSL installation, the option was implicitly changed to the bundled value and yaSSL was used instead. Now CMake exits with an error so the user knows that the option value must be changed. (Bug #69744, Bug #17162055)
- The DBUG_ENTER string for the THD::increment_questions_counter() function incorrectly named the THD::increment_updates_counter() function. (Bug #69989, Bug #17297266)
- RPM packages did not provide lowercase tags for their contents. For example, a server RPM indicated that it provided MySQL-server, but not mysql-server. (Bug #69830, Bug #17211588)
- mysql.h no longer defines __WIN__ on Windows, and the MySQL sources have been changed to test for _WIN32 instead. (Bug #20338, Bug #11745828)

Changes in MySQL 5.7.1 (2013-04-23, Milestone 11)

Note

This is a milestone release, for use at your own risk. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running mysql_upgrade. For example, you may find it necessary to dump your data with mysqldump before the upgrade and reload it afterward.

Audit Log Plugin Notes

- Several changes were made to the audit log plugin for better compatibility with Oracle Audit Vault.

  The format of the audit log file has changed:

  - Information within <AUDIT_RECORD> elements written in the old format using attributes is written in the new format using subelements.
  - The new format includes more information in <AUDIT_RECORD> elements. Every element includes a RECORD_ID value providing a unique identifier. The TIMESTAMP value includes time zone information. Query records include HOST, IP, OS_LOGIN, and USER information, as well as COMMAND_CLASS and STATUS_CODE values.

    The STATUS_CODE value differs from the existing STATUS value: STATUS_CODE is 0 for success and 1 for error, which is compatible with the EZ_collector consumer for Audit Vault. STATUS is the value of the mysql_errno() C API function. This is 0 for success and nonzero for error, and thus is not necessarily 1 for error.

  Example of old <AUDIT_RECORD> format:

```
<AUDIT_RECORD
 TIMESTAMP="2013-04-15T15:27:27" NAME="Query" CONNECTION_ID="3" STATUS="0" SQLTEXT="SELECT 1" />
```

Example of new `<AUDIT_RECORD>` format:

```
<AUDIT_RECORD>
 <TIMESTAMP>2013-04-15T15:27:27 UTC</TIMESTAMP>
 <RECORD_ID>3998_2013-04-15T15:27:27</RECORD_ID>
 <NAME>Query</NAME>
 <CONNECTION_ID>3</CONNECTION_ID>
 <STATUS>0</STATUS>
 <STATUS_CODE>0</STATUS_CODE>
 <USER>root[root] @ localhost [127.0.0.1]</USER>
 <OS_LOGIN></OS_LOGIN>
 <HOST>localhost</HOST>
 <IP>127.0.0.1</IP>
 <COMMAND_CLASS>select</COMMAND_CLASS>
 <SQLTEXT>SELECT 1</SQLTEXT>
</AUDIT_RECORD>
```

When the audit log plugin rotates the audit log file, it uses a different file name format. For a log file named `audit.log`, the plugin previously renamed the file to `audit.log.`*`TIMESTAMP`*. The plugin now renames the file to `audit.log.`*`TIMESTAMP`*`.xml` to indicate that it is an XML file.

For information about the audit log plugin, see MySQL Enterprise Audit Log Plugin.

If you previously used an older version of the audit log plugin, use this procedure to avoid writing new-format log entries to an existing log file that contains old-format entries:

1. Stop the server.
2. Rename the current audit log file manually. This file will contain only old-format log entries.
3. Update the server and restart it. The audit log plugin will create a new log file, which will contain only new-format log entries.

The API for writing audit plugins has also changed. The `mysql_event_general` structure has new members to represent client host name and IP address, command class, and external user. For more information, see Writing Audit Plugins.

Functionality Added or Changed

- **Performance:** String hashing overhead was reduced. This also improves performance for metadata locking, the table definition cache, and Performance Schema table I/O and file I/O instrumentation. (Bug #13944392)
- **Incompatible Change:** `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` output used a mix of `row_count` and `count` attributes. These are now all `count`. Similarly, the output used a mix of `row_size` and `size` attributes. These are now all `size`. (Bug #16165468)
- **InnoDB:** Online index renaming is supported by `ALTER TABLE`, which now includes a `RENAME INDEX` clause, as shown in the following example: “`ALTER TABLE t RENAME INDEX i1 TO i2`”, where `i1` is the current name of the index and `i2` is the new name.

  The result of “`ALTER TABLE t RENAME INDEX i1 TO i2`” would be a table with contents and structure that is identical to the old version of “t1” except for the index name, which is now “i2” instead of “i1”.

- **InnoDB:** VARCHAR size may be increased using an in-place ALTER TABLE, as in this example:

  ```
  ALTER TABLE t1 ALGORITHM=INPLACE, CHANGE COLUMN c1 c1 VARCHAR(255);
  ```

  This is true as long as the number of length bytes required by a VARCHAR column remains the same. For VARCHAR values of 0 to 255, one length byte is required to encode the value. For VARCHAR values of 256 bytes or more, two length bytes are required. As a result, in-place ALTER TABLE only supports increasing VARCHAR size from 0 to 255 bytes or increasing VARCHAR size from a value equal to or greater than 256 bytes. In-place ALTER TABLE does not support increasing VARCHAR size from less than 256 bytes to a value equal to or greater than 256 bytes. In this case, the number of required length bytes would change from 1 to 2, which is only supported by a table copy (ALGORITHM=COPY).

  Decreasing VARCHAR size using in-place ALTER TABLE is not supported. Decreasing VARCHAR size requires a table copy (ALGORITHM=COPY).

  For additional details, refer to the “Column Properties” information in Overview of Online DDL.

- **InnoDB:** DDL performance for InnoDB temporary tables is improved through optimization of CREATE TABLE, DROP TABLE, TRUNCATE TABLE, and ALTER TABLE statements. Optimizations were achieved by limiting actions performed by DDL statements to only those necessary for temporary tables.

- **InnoDB:** InnoDB temporary table metadata is no longer stored to InnoDB system tables. Instead, a new table, INNODB_TEMP_TABLE_INFO, provides users with a snapshot of active temporary tables. The table contains metadata and reports on all user and system-created temporary tables that are active within a given InnoDB instance. The table is created when the first SELECT statement is run against it.

- **InnoDB:** Added support for spatial data types currently supported by MySQL including POINT, LINE_STRING, POLYGON, MULTI_POINT, MULTI_POLYGON, MULTI_LINE_STRING, GEOMETRY_COLLECTION, and GEOMETRY.

  Prior to this release, InnoDB would store spatial data as binary BLOB data. BLOB remains the underlying data type but spatial data types are now mapped to a new InnoDB internal data type, DATA_GEOMETRY. With BLOB as the underlying data type, a prefix index can still be used on the GEOMETRY data column.

- **InnoDB:** Added a separate tablespace for all non-compressed InnoDB temporary tables. The new tablespace is always recreated on server startup.

  The new tablespace, ibtmp1, is located in the MySQL data directory (datadir) by default. A newly added configuration file option, innodb_temp_data_file_path, allows for a user-defined temporary data file path. For related information, see InnoDB Temporary Table Undo Logs.

- **Partitioning:** HANDLER statements are now supported with partitioned tables.

- **Replication:** An Auto_Position column has been added to the output generated by SHOW SLAVE STATUS. The value of this column shows whether replication autopositioning is in use. If autopositioning is enabled—that is, if MASTER_AUTO_POSITION = 1 was set by the last successful CHANGE MASTER TO statement that was executed on the slave—then the column's value is 1; if not, then the value is 0. (Bug #15992220)

- **Replication:** Added the --rewrite-db option for mysqlbinlog, which allows mysqlbinlog to rewrite the names of databases when playing back binary logs written using the row-based logging format. Multiple rewrite rules can be created by specifying the option multiple times.

- **Replication:** The functions GTID_SUBTRACT() and GTID_SUBSET() were formerly available in libmysqld only when it was built with replication support. Now these functions are always available when using this library, regardless of how it was built.

- MySQL no longer uses the default OpenSSL compression. (Bug #16235681)
- There is now a distinct error code (ER_MUST_CHANGE_PASSWORD_LOGIN) for the error sent by the server to a client authenticating with an expired password. (Bug #16102943)
- In RPM packages built for Unbreakable Linux Network, libmysqld.so now has a version number. (Bug #15972480)
- Error messages for ALTER TABLE statement using a LOCK or ALGORITHM value not supported for the given operation were very generic. The server now produces more informative messages. (Bug #15902911)
- If a client with an expired password connected but old_passwords was not the value required to select the password hashing format appropriate for the client account, there was no way for the client to determine the proper value. Now the server automatically sets the session old_passwords value appropriately for the account authentication method. For example, if the account uses the sha256_password authentication plugin, the server sets old_passwords=2. (Bug #15892194)
- mysql_config_editor now supports --port and --socket options for specifying TCP/IP port number and Unix socket file name. (Bug #15851247)
- mysqldump now supports an --ignore-error option. The option value is a comma-separated list of error numbers specifying the errors to ignore during mysqldump execution. If the --force option is also given to ignore all errors, --force takes precedence. (Bug #15855723)
- mysqlcheck has a new --skip-database option. The option value is the name of a database (case sensitive) for which checks should be skipped.

  mysql_upgrade uses this option to upgrade the system tables in the mysql database before tables in other databases: It upgrade the mysql database, then all databases except the mysql database. This avoids problems that can occur if user tables are upgraded before the system tables. (Bug #14697538)
- The validate_password_policy_number system variable was renamed to validate_password_policy. (Bug #14588121)
- Previously, on Linux the server failed to perform stack backtrace attempts for versions of glibc older than the current minimum supported version (2.3). Now on such attempts the server displays a message that the glibc version is too old to support backtracing. (Bug #14475946)
- In JSON-format EXPLAIN output, the attached_condition information for subqueries now includes select# to indicate the relative order of subquery execution. (Bug #13897507)
- The following changes were made to the sandbox mode that the server uses to handle client connections for accounts with expired passwords:
  - There is a new disconnect_on_expired_password system variable (default: enabled). This controls how the server treats expired-password accounts.
  - Two flags were added to the C API client library: MYSQL_OPT_CAN_HANDLE_EXPIRED_PASSWORDS for mysql_options() and CLIENT_CAN_HANDLE_EXPIRED_PASSWORDS for mysql_real_connect(). Each flag enables a client program to indicate whether it can handle sandbox mode for accounts with expired passwords.

    MYSQL_OPT_CAN_HANDLE_EXPIRED_PASSWORDS is enabled for mysqltest unconditionally, for mysql in interactive mode, and for mysqladmin if the first command is password.

For more information about how the client-side flags interact with disconnect_on_expired_password, see Password Expiration and Sandbox Mode. (Bug #67568, Bug #15874023)

- If a user attempted to access a nonexistent column for which the user had no access, the server returned an error indicating that the column did not exist. Now the server returns an error indicating that the user does not have privileges for the column, which provides no information about column existence. (Bug #19947, Bug #11745788)

- The deprecated innodb_mirrored_log_groups system variable has been removed.

- The mysql client now has a --syslog option that causes interactive statements to be sent to the system syslog facility. Logging is suppressed for statements that match the default "ignore" pattern list ("*IDENTIFIED*:*PASSWORD*"), as well as statements that match any patterns specified using the --histignore option. For more information, see mysql Logging.

- ALTER TABLE now supports a RENAME INDEX clause that renames an index. The change is made in place without a table-copy operation.

- The MySQL test suite mysql-test-run.sh program now starts the server with InnoDB rather than MyISAM as the default storage engine. To maintain compatibility of test results with existing result files, test cases were modified to add a line that includes the force_default_myisam.inc file as necessary. In a future release, for those test cases not specifically requiring MyISAM, that line will be removed (so they run with InnoDB) and test results will be updated.

Bugs Fixed

- **Performance; InnoDB:** The DROP TABLE statement for a table using compression could be slower than necessary, causing a stall for several seconds. MySQL was unnecessarily decompressing pages in the buffer pool related to the table as part of the DROP operation. (Bug #16067973)

- **Incompatible Change; Partitioning:** Changes in the KEY partitioning hashing functions used with numeric, date and time, ENUM, and SET columns in MySQL 5.5 makes tables using partitioning or subpartitioning by KEY on any of the affected column types and created on a MySQL 5.5 or later server incompatible with a MySQL 5.1 server. This is because the partition IDs as calculated by a MySQL 5.5 or later server almost certainly differ from those calculated by a MySQL 5.1 server for the same table definition and data as a result of the changes in these functions.

  The principal changes in the KEY partitioning implementation in MySQL 5.5 resulting in this issue were as follows: 1. The hash function used for numeric and date and time columns changed from binary to character-based. 2. The base used for hashing of ENUM and SET columns changed from latin1 ci characters to binary.

  The fix involves adding the capability in MySQL 5.5 and later to choose which type of hashing to use for KEY partitioning, which is implemented with a new ALGORITHM extension to the PARTITION BY KEY option for CREATE TABLE and ALTER TABLE. Specifying PARTITION BY KEY ALGORITHM=1 ([*columns*]) causes the server to use the hashing functions as implemented in MySQL 5.1; using ALGORITHM=2 causes the server to use the hashing functions from MySQL 5.5 and later. ALGORITHM=2 is the default. Using the appropriate value for ALGORITHM, you can perform any of the following tasks:

  - Create KEY partitioned tables in MySQL 5.5 and later that are compatible with MySQL 5.1, using CREATE TABLE ... PARTITION BY KEY ALGORITHM=1 (...).

  - Downgrade KEY partitioned tables that were created in MySQL 5.5 or later to become compatible with MySQL 5.1, using ALTER TABLE ... PARTITION BY KEY ALGORITHM=1 (...).

- Upgrade KEY partitioned tables originally created in MySQL 5.1 to use hashing as in MySQL 5.5 and later, using ALTER TABLE ... PARTITION BY KEY ALGORITHM=2 (...).

  Important: After such tables are upgraded, they cannot be used any longer with MySQL 5.1 unless they are first downgraded again using ALTER TABLE ... PARTITION BY KEY ALGORITHM=1 (...) on a MySQL server supporting this option.

This syntax is not backward compatible, and causes errors in older versions of the MySQL server. When generating CREATE TABLE ... PARTITION BY KEY statements, mysqldump brackets any occurrence of ALGORITHM=1 or ALGORITHM=2 in conditional comments such that it is ignored by a MySQL server whose version is not at least 5.5.31. An additional consideration for upgrades is that MySQL 5.6 servers prior to MySQL 5.6.11 do not ignore the ALGORITHM option in such statements when generated by a MySQL 5.5 server, due to the that the conditional comments refer to version 5.5.31; in this case, you must edit the dump manually and remove or comment out the option wherever it occurs before attempting to load it into a MySQL 5.6.10 or earlier MySQL 5.6 server. This is not an issue for dumps generated by MySQL 5.6.11 or later version of mysqldump, where the version used in such comments is 5.6.11. For more information, see ALTER TABLE Partition Operations.

As part of this fix, a spurious assertion by InnoDB that a deleted row had previously been read, causing the server to assert on delete of a row that the row was in the wrong partition, was also removed. (Bug #14521864, Bug #66462, Bug #16093958, Bug #16274455)

References: See also Bug #11759782.

- **Important Change; Plugin API; Replication:** Because the behavior of the fulltext plugin may vary between MySQL servers, it is not possible to guarantee that statements using this plugin produce the same results on masters and slaves. For this reason, statements depending on the fulltext plugin are now marked as unsafe for statement-based logging. This means that such statements are logged using row format when binlog_format=MIXED, and cause a warning to be generated when binlog_format=STATEMENT. (Bug #11756280, Bug #48183)

- **Important Change; Replication**

  Important

  This fix was reverted in MySQL 5.7.2. See Changes in MySQL 5.7.2 (2013-09-21, Milestone 12).

  Executing a statement that performs an implicit commit but whose changes are not logged when gtid_next is set to any value other than AUTOMATIC is not permitted. Now in such cases, the statement fails with an error. This includes the statements in the following list:

  - CHANGE MASTER TO
  - START SLAVE
  - STOP SLAVE
  - REPAIR TABLE
  - OPTIMIZE TABLE
  - ANALYZE TABLE
  - CHECK TABLE

  - CREATE SERVER
  - ALTER SERVER
  - DROP SERVER
  - CACHE INDEX
  - LOAD INDEX INTO CACHE
  - FLUSH
  - RESET

  (Bug #16062608)

  References: See also Bug #16484323.

- **Important Change; Replication:** The version number reported by mysqlbinlog --version has been increased to 3.4. (Bug #15894381, Bug #67643)

- **Important Change; Replication:** The lettercasing used for displaying UUIDs in global transaction identifiers was inconsistent. Now, all GTID values use lowercase, including those shown in the Retrieved_Gtid_Set and Executed_Gtid_Set columns from the output of SHOW SLAVE STATUS. (Bug #15869441)

- **Important Note; Replication:** Using row-based logging to replicate from a table to a same-named view led to a failure on the slave. Now, when using row-based logging, the target object type is checked prior to performing any DML, and an error is given if the target on the slave is not actually a table.

  Note

  It remains possible to replicate from a table to a same-named view using statement-based logging.

  (Bug #11752707, Bug #43975)

- **MySQL Cluster:** The setting for the DefaultOperationRedoProblemAction API node configuration parameter was ignored, and the default value used instead. (Bug #15855588)

- **MySQL Cluster:** The recently added LCP fragment scan watchdog occasionally reported problems with LCP fragment scans having very high table id, fragment id, and row count values.

  This was due to the watchdog not accounting for the time spent draining the backup buffer used to buffer rows before writing to the fragment checkpoint file.

  Now, in the final stage of an LCP fragment scan, the watchdog switches from monitoring rows scanned to monitoring the buffer size in bytes. The buffer size should decrease as data is written to the file, after which the file should be promptly closed. (Bug #14680057)

- **MySQL Cluster:** Job buffers act as the internal queues for work requests (signals) between block threads in ndbmtd and could be exhausted if too many signals are sent to a block thread.

  Performing pushed joins in the DBSPJ kernel block can execute multiple branches of the query tree in parallel, which means that the number of signals being sent can increase as more branches are executed. If DBSPJ execution cannot be completed before the job buffers are filled, the data node can fail.

This problem could be identified by multiple instances of the message sleeploop 10!! in the cluster out log, possibly followed by job buffer full. If the job buffers overflowed more gradually, there could also be failures due to error 1205 (Lock wait timeout exceeded), shutdowns initiated by the watchdog timer, or other timeout related errors. These were due to the slowdown caused by the 'sleeploop'.

Normally up to a 1:4 fanout ratio between consumed and produced signals is permitted. However, since there can be a potentially unlimited number of rows returned from the scan (and multiple scans of this type executing in parallel), any ratio greater 1:1 in such cases makes it possible to overflow the job buffers.

The fix for this issue defers any lookup child which otherwise would have been executed in parallel with another is deferred, to resume when its parallel child completes one of its own requests. This restricts the fanout ratio for bushy scan-lookup joins to 1:1. (Bug #14709490)

References: See also Bug #14648712.

- **InnoDB:** When parsing a delimited search string such as “abc-def” in a full-text search, InnoDB now uses the same word delimiters as MyISAM. (Bug #16419661)
- **InnoDB:** This fix improves code readability by addressing naming inconsistencies for InnoDB PERFORMANCE_SCHEMA key declarations. (Bug #16414044)
- **InnoDB:** This fix disables a condition for extra splitting of clustered index leaf pages, on compressed tables. Extra page splitting was only done to reserve space for future updates, so that future page splits could be avoided. (Bug #16401801)
- **InnoDB:** For InnoDB tables, if a PRIMARY KEY on a VARCHAR column (or prefix) was empty, index page compression could fail. (Bug #16400920)
- **InnoDB:** Status values in the innodb_ft_config table would not update. The innodb_ft_config is intended for internal configuration and should not be used for statistical information purposes. To avoid confusion, column values that are intended for internal use have been removed from the innodb_ft_config table. This fix also removes the innodb_ft_config table and other internal full text search-related tables that were unintentionally exposed. (Bug #16409494, Bug #68502)
- **InnoDB:** With innodb_api_enable_mdl=OFF, an ALTER TABLE operation on an InnoDB table that required a table copy could cause a server exit. (Bug #16287411)
- **InnoDB:** Improper testing of compatibility between the referencing and referenced during ALTER TABLE ... ADD FOREIGN key could cause a server exit. (Bug #16330036)
- **InnoDB:** Rollback did not include changes made to temporary tables by read-only transactions. (Bug #16310467)
- **InnoDB:** The InnoDB page-splitting algorithm could recurse excessively. (Bug #16345265)
- **InnoDB:** For debug builds, InnoDB status exporting was subject to a race condition that could cause a server exit. (Bug #16292043)
- **InnoDB:** When using ALTER TABLE to set an AUTO_INCREMENT column value to a user-specified value, InnoDB would set the AUTO_INCREMENT value to the user-specified value even when the AUTO_INCREMENT value is greater than the user-specified value. This fix ensures that the AUTO_INCREMENT value is set to the maximum of the user-specified value and MAX(auto_increment_column)+1, which is the expected behaviour. (Bug #16310273)

- **InnoDB:** Importing a tablespace with the configuration file present would not import the data file. This problem would occur when all pages are not flushed from the buffer pool after a table is altered using the copy and rename approach. This fix ensures that all pages are flushed from the buffer pool when a table is altered using the copy and rename approach. (Bug #16318052)
- **InnoDB:** The `lock_validate` function, which is only present in debug builds, acquired and released mutexes to avoid hogging them. This behavior introduced a window wherein changes to the hash table could occur while code traversed the same set of data. This fix updates `lock_validate` logic to collect all records for which locks must be validated, releases mutexes, and runs a loop to validate record locks. (Bug #16235056)
- **InnoDB:** `ALTER TABLE` functions would perform a check to see if InnoDB is in read-only mode (`srv_read_only_mode=true`). If InnoDB was in read-only mode, the check would return a successful status and do nothing else. This fix replaces `srv_read_only_mode` check conditions with debug assertions. (Bug #16227539)
- **InnoDB:** InnoDB now aborts execution on Windows by calling the `abort()` function directly, as it does on other platforms. (Bug #16263506)
- **InnoDB:** When the InnoDB buffer pool is almost filled with 4KB compressed pages, inserting into 16KB compact tables would cause 8KB `pages_free` to increase, which could potentially slow or stall inserts. (Bug #16223169)
- **InnoDB:** An assertion failure would occur in `heap->magic_n == MEM_BLOCK_MAGIC_N` due to a race condition that appeared when `row_merge_read_clustered_index()` returned an error. (Bug #16275237)
- **InnoDB:** This fix removes an unnecessary debug assertion related to page_hash locks which only affects debug builds. The debug assertion is no longer valid and should have been removed when hash_lock array was introduced in MySQL 5.6. (Bug #16263167)
- **InnoDB:** Without warning, InnoDB would silently set `innodb-buffer-pool-instances` to 1 if the buffer pool size is less than 1GB. For example, if `innodb-buffer-pool-size` is set to 200M and `innodb-buffer-pool-instances` is set to 4, InnoDB would silently set `innodb-buffer-pool-instances` to 1. This fix implements a warning message and new logic for `innodb-buffer-pool-size` and `innodb-buffer-pool-instances`. (Bug #16249500, Bug #61239)
- **InnoDB:** When the primary key of a table includes a column prefix, and a full-text index is defined on the table, a full-text search resulted in an unnecessary warning being written to the error log. This fix suppresses the unnecessary warning. (Bug #16169411)
- **InnoDB:** When InnoDB locking code was revised, a call to register lock waits was inadvertently removed. This fix adds the call back to the InnoDB locking code. (Bug #16208201)
- **InnoDB:** A direct call to the `trx_start_if_not_started_xa_low()` function would cause a debug assertion. (Bug #16178995)
- **InnoDB:** In the case of LOCK WAIT for an insert in a foreign key table, `InnoDB` could report a false dictionary-changed error and cause the insert to fail rather than being retried. (Bug #16174255)
- **InnoDB:** In some cases, deadlock detection did not work, resulting in sessions hanging waiting for a lock-wait timeout. (Bug #16169638)
- **InnoDB:** An in-place `ALTER TABLE` on an `InnoDB` table could fail to delete the statistics for the old primary key from the `mysql.innodb_index_stats` table. (Bug #16170451)

- **InnoDB:** This fix updates InnoDB code in ha_innodb.cc and handler0alter.cc to use TABLE::key_info instead of both TABLE::key_info and TABLE_SHARE::key_info. (Bug #16215361)

- **InnoDB:** Arithmetic underflow during page compression for CREATE TABLE on an InnoDB table could cause a server exit. (Bug #16089381)

- **InnoDB:** LOCK_TIME would not be logged correctly in the slow query log. LOCK_TIME did not account for InnoDB row lock wait time. (Bug #16097753)

- **InnoDB:** For debug builds, online ALTER TABLE operations for InnoDB tables could cause a server exit during table rebuilding. (Bug #16063835)

- **InnoDB:** In some cases, the InnoDB purge coordinator did not use all available purge threads, resulting in suboptimal purge activity. (Bug #16037372)

- **InnoDB:** ALTER TABLE for InnoDB tables was not fully atomic. (Bug #15989081)

- **InnoDB:** This fix replaces most uses of UT_SORT_FUNCTION_BODY, an InnoDB recursive merge sort, with the std::sort() function from the C++ Standard Template Library (STL). The std::sort() function requires less memory and is faster due to in-line execution. (Bug #15920744)

- **InnoDB:** This fix implements a 256-byte boundary for extending a VARCHAR column instead of 256-character boundary. This change allows for in-place extension of a VARCHAR column through an update of the data dictionary. (Bug #15863023)

- **InnoDB:** This fix addresses unnecessary buffer pool lookups that would occur while freeing blob pages, and implements a debug status instrument, innodb_ahi_drop_lookups, for testing purposes. (Bug #15866009)

- **InnoDB:** Creating numerous tables, each with a full-text search index, could result in excessive memory consumption. This bug fix adds a new configuration parameter, innodb_ft_total_cache_size, which defines a global memory limit for full-text search indexes. If the global limit is reached by an index operation, a force sync is triggered. (Bug #14834698, Bug #16817453)

- **InnoDB:** This fix modifies InnoDB code to ensure that unused thread handles are closed when the thread exits, instead of leaving thread handles open until shutdown of mysqld on Windows. (Bug #14762796)

- **InnoDB:** This fix removes unnecessary overhead by removing table locking and disabling read view creation and MVCC when InnoDB is started in read-only mode (--innodb-read-only=true). (Bug #14729365)

- **InnoDB:** A regression introduced by the fix for Bug#14100254 would result in a “!BPAGE->FILE_PAGE_WAS_FREED” assertion. (Bug #14676249)

- **InnoDB:** An error at the filesystem level, such as too many open files, could cause an unhandled error during an ALTER TABLE operation. The error could be accompanied by Valgrind warnings, and by this assertion message:

  ```
  Assertion `! is_set()' failed.
  mysqld got signal 6 ;
  ```

  (Bug #14628410, Bug #16000909)

- **InnoDB:** The INNODB_SYNC_ARRAY_SIZE variable was incorrectly allowed to be configured at runtime. As documented, INNODB_SYNC_ARRAY_SIZE must be configured when the MySQL instance is starting up, and cannot be changed afterward. This fix changes INNODB_SYNC_ARRAY_SIZE to a non-dynamic variable, as intended. (Bug #14629979)

- **InnoDB:** The server could exit during an attempt by InnoDB to reorganize or compress a compressed secondary index page. (Bug #14606334)
- **InnoDB:** Full-text search (FTS) index savepoint information would not be set resulting in a severe error when attempting to rollback to the savepoint. (Bug #14639605, Bug #17456092)
- **InnoDB:** A DML operation performed while a RENAME TABLE operation waits for pending I/O operations on the tablespace to complete would result in a deadlock. (Bug #14556349)
- **InnoDB:** Attempting to unninstall the InnoDB memcached Plugin while the plugin is still installing caused the Mysql server to terminate. While the plugin deamon thread was still initializing, plugin variables were not yet set and the uninstall process could not cleanup resources. This fix adds a variable to indicate initialization status. If initialization is incomplete, the uninstall process will wait. (Bug #14279541)
- **InnoDB:** If the value of innodb_force_recovery was less than 6, opening a corrupted table might loop forever if a corrupted page was read when calculating statistics for the table. Information about the corrupted page was written repeatedly to the error log, possibly causing a disk space issue. The fix causes the server to halt after a fixed number of failed attempts to read the page. To troubleshoot such a corruption issue, set innodb_force_recovery=6 and restart. (Bug #14147491, Bug #65469)
- **InnoDB:** When printing out long semaphore wait diagnostics, sync_array_cell_print() ran into a segmentation violation (SEGV) caused by a race condition. This fix addresses the race condition by allowing the cell to be freed while it is being printed. (Bug #13997024)
- **InnoDB:** Killing a query caused an InnoDB assertion failure when the same table (cursor) instance was used again. This is the result of a regression error introduced by the fix for Bug#14704286. The fix introduced a check to handle kill signals for long running queries but the cursor was not restored to the proper state. (Bug #68051, Bug #16088883)
- **InnoDB:** On startup, InnoDB reported a message on 64-bit Linux and 64-bit Windows systems stating that the CPU does not support crc32 instructions. On Windows, InnoDB does not use crc32 instructions even if supported by the CPU. This fix revises the wording of the message and implements a check for availability of crc32 instructions. (Bug #68035, Bug #16075806)
- **InnoDB:** The length of internally generated foreign key names was not checked. If internally generated foreign key names were over the 64 character limit, this resulted in invalid DDL from SHOW CREATE TABLE. This fix checks the length of internally generated foreign key names and reports an error message if the limit is exceeded. (Bug #44541, Bug #11753153)
- **InnoDB:** This fix removes left-over prototype code for srv_parse_log_group_home_dirs, and related header comments. (Bug #68133, Bug #16198764)
- **InnoDB:** Attempting to replace the default InnoDB full-text search (FTS) stopword list by creating an InnoDB table with the same structure as INFORMATION_SCHEMA.innodb_ft_default_stopword would result in an error. SHOW CREATE TABLE revealed that the new InnoDB table was created with CHARSET=utf8. The InnoDB FTS stopword table validity check only supported latin1. This fix extends the validity check for all supported character sets. (Bug #68450, Bug #16373868)
- **Partitioning:** A query on a table partitioned by range and using TO_DAYS() as a partitioing function always included the first partition of the table when pruning. This happened regardless of the range employed in the BETWEEN clause of such a query. (Bug #15843818, Bug #49754)
- **Partitioning:** Partition pruning is now enabled for tables using a storage engine that provides automatic partitioning, such as the NDB storage engine, but which are explicitly partitioned. Previously, pruning

was disabled for all tables using such a storage engine, whether or not the tables had explicitly defined partitions.

In addition, as part of this fix, explicit partition selection is now disabled for tables using a storage engine (such as `NDB`) that provides automatic partitioning. (Bug #14827952)

References: See also Bug #14672885.

- **Partitioning:** Execution of `ALTER TABLE ... DROP PARTITION` against a view caused the server to crash, rather than fail with an error as expected. (Bug #14653504)
- **Partitioning:** A query result was not sorted if both `DISTINCT` and `ORDER BY` were used and the underlying table was partitioned. (Bug #14058167)
- **Partitioning:** Inserting any number of rows into an `ARCHIVE` table that used more than 1000 partitions and then attempting to drop the table caused the MySQL Server to fail. (Bug #13819630, Bug #64580)
- **Replication; Linux; Microsoft Windows:** Replication failed between a Linux master using `lower_case_table_names` set to 0 and a Windows slave having `lower_case_table_names` set to 2, after a replicated table was opened on the slave; in addition, `FLUSH TABLES` was required afterwards to see which updates had actually been applied on the slave. This was because `lower_case_table_names` was checked only to see whether it was equal to 1 prior to forcing a conversion of replicated database object names to lower case for checking the table cache. Now in such cases, `lower_case_table_names` is checked to see whether it is set to a nonzero value. (Bug #16061982)
- **Replication:** When using GTIDs and binary log auto-positioning, the master had to scan all binary logs whenever the slave reconnected (due to reasons such as I/O thread failure or a change of master) before it could send any events to slave. Now, the master starts from the oldest binary log that contains any GTID not found on the slave. (Bug #16340322, Bug #68386)
- **Replication:** When the server version of the master was greater than or equal to 10, replication to a slave having a lower server version failed. (Bug #16237051, Bug #68187)
- **Replication:** When replicating to a MySQL 5.6 master to an older slave, Error 1193 (**`ER_UNKNOWN_SYSTEM_VARIABLE`**) was logged with a message such as `Unknown system variable 'SERVER_UUID' on master, maybe it is a *VERY OLD MASTER*`. This message has been improved to include more information, similar to this one: `Unknown system variable 'SERVER_UUID' on master. A probable cause is that the variable is not supported on the master (version: 5.5.31), even though it is on the slave (version: 5.6.11)`. (Bug #16216404, Bug #68164)
- **Replication:** A zero-length name for a user variable (such as ``@` ` ``) was incorrectly considered to be a sign of data or network corruption when reading from the binary log. (Bug #16200555, Bug #68135)
- **Replication:** Running `SHOW RELAYLOG EVENTS` at a slave where no relay log file is present returned the following incorrect error message: "`Error when executing command SHOW BINLOG EVENTS: Could not find target log.`" The error message text has been changed to: "`Error when executing command SHOW RELAYLOG EVENTS: Could not find target log.`" (Bug #16191895)
- **Replication:** When MTS is on and transactions are being applied, the slave coordinator would hang when encountering a checksum error on a transaction event. This was due to a deadlock situation in which the coordinator assumed a normal stop while a worker waited for the coordinator to dispatch more events. For debug builds, the problem appeared as an assertion failure, which was due to the coordinator not setting `thd->is_error()` when encountering an error. (Bug #16210351)

- **Replication:** The print format specifier for the server_id was incorrectly defined as a signed 32-bit integer with a range of -2144783647 to 2144783648. This fix changes the server_id integer type to an unsigned 32-bit integer type, with a range of 0 to 4294967295, which is the documented range for the --server-id option. (Bug #16210894)

- **Replication:** mysqlbinlog can connect to a remote server and read its binary logs. In MySQL 5.6 and later, this tool can also wait for the server to generate and send additional events, in practice behaving like a slave connecting to a master. In cases where the server sent a heartbeat, mysqlbinlog was unable to handle it properly. As a consequence, mysqlbinlog failed at this point, without reading any more events from the server. To fix this problem, mysqlbinlog now ignores any binary log events of type HEARTBEAT_LOG_EVENT that it receives. (Bug #16104206)

- **Replication:** STOP SLAVE could cause a deadlock when issued concurrently with a statement such as SHOW STATUS that retrieved the values for one or more of the status variables Slave_retried_transactions, Slave_heartbeat_period, Slave_received_heartbeats, Slave_last_heartbeat, or Slave_running. (Bug #16088188, Bug #67545)

  References: See also Bug #16088114.

- **Replication:** Using the --replicate-* options (see Replication Slave Options and Variables) could in some cases lead to a memory leak on the slave. (Bug #16056813, Bug #67983)

- **Replication:** Backtick (`) characters were not always handled correctly in internally generated SQL statements, which could sometimes lead to errors on the slave. (Bug #16084594, Bug #68045)

  References: This bug is a regression of Bug #14548159, Bug #66550.

- **Replication:** The session-level value for gtid_next was incorrectly reset on the slave for all rollbacks, which meant that GTIDs could be lost for multi-statement transactions, causing the slave to stop with an **ER_GTID_NEXT_TYPE_UNDEFINED_GROUP** error. Now this is done only when a complete transaction is being rolled back, or when autocommit is enabled. (Bug #16084206)

- **Replication:** In order to provision or to restore a server using GTIDs, it is possible to set gtid_purged to a given GTID set listing the transactions that were imported. This operation requires that the global gtid_executed and gtid_purged server system variables are empty. (This is done in order to avoid the possibility of overriding server-generated GTIDs.)

  The error message GTID_PURGED can only be set when GTID_EXECUTED is empty that was raised when this requirement was not met could be confusing or misleading because it did not specify the scope of the affected variables. To prevent this from happening, error messages that refer to variables relating to GTIDs now specify the scope of any such variables when they do so. (Bug #16084426, Bug #68038)

- **Replication:** In certain cases, the dump thread could send a heartbeat out of synchronisation with format description events. One of the effects of this issue what that, after provisioning a new server from a backup data directory and setting --gtid-mode=ON and enabling autopositioning (see CHANGE MASTER TO Syntax), replication failed to start, with the error Read invalid event from master.... The same problem could also cause GTID-based replication to fail due to skipped events following a unplanned shutdown of the master. (Bug #16051857)

- **Replication:** In some cases, when the slave could not recognize the server version of the master, this could cause the slave to fail. (Bug #16056365)

- **Replication:** Dropping a table that includes non-regular ASCII characters in the table name caused a replication failure. The parser converted the table name into standard charset characters and stored the table name in the table_name variable. When the drop table query was regenerated using the table_name variable, the table name was not converted back to the original charset.

Additionally, table and database names with 64 characters caused an assert failure. The assert required the table or database name to be less than 128 characters. Latin characters require two-bytes each, which requires an assert condition of less than or equal to 128 bits.

The fix includes a new function to convert tables names back to the original charset, and a correction to the assert condition allowing table and database names be less than or equal to 128 bits. (Bug #16066637)

- **Replication:** Replication failed when a replicated `LOAD DATA` statement inserted rows into a view. (Bug #15993712, Bug #67878)
- **Replication:** When using GTID-based replication, and whenever a transaction was executed on the master but was not sent to the slave because the slave already had a transaction with that ID, semisynchrononous replication timed out. One case in which this could happen was during a failover operation where the new master started behind the new slave. (Bug #15985893)
- **Replication:** An unnecessary flush to disk performed after every transaction when using `FILE` as the replication info repository type could degrade performance. Now this is done only when both data and relay log info is stored in (transactional) tables. (Bug #15980626)
- **Replication:** When a slave read a table map event from the binary log, it assumed that the metadata size was always less than twice the column count of the table in use, which failed when the event contained the wrong value for this field. (Bug #15830022)
- **Replication:** When reading row log events from the binary log, the slave assumed that these events were always valid; because of this, an event having an invalid binary log offset could cause the slave to crash. Now in such cases, the slave fails gracefully, and an error is reported, if any of the fields in a given row event are invalid. (Bug #15829568)
- **Replication:** Table IDs used in replication were defined as type `ulong` on the master and `uint` on the slave. In addition, the maximum value for table IDs in binary log events is 6 bytes (281474976710655). This combination of factors led to the following issues:

  - Data could be lost on the slave when a table was assigned an ID greater than `uint`.
  - Table IDs greater than 281474976710655 were written to the binary log as 281474976710655.
  - This led to a stopped slave when the slave encountered two tables having the same table ID.

  To fix these problems, IDs are now defined by both master and slave as type `ulonglong` but constrained to a range of 0 to 281474976710655, restarting from 0 when it exceeds this value. (Bug #14801955, Bug #67352)
- **Replication:** `MASTER_POS_WAIT()` could hang or return -1 due to invalid updates by the slave SQL thread when transactions were skipped by the GTID protocol. (Bug #14737388)

  References: See also Bug #15927032.
- **Replication:** Trying to execute a Stop event on a multi-threaded slave could cause unwanted updates to the relay log, leading the slave to lose synchronization with the master. (Bug #14737388)
- **Replication:** Internal objects used for relay log information were only partially deleted before freeing their memory. (Bug #14677824)
- **Replication:** When the server starts, it checks whether semisynchronous replication has been enabled without a lock, and, if so, it takes the lock, then tests again. Disabling semisynchronous replication

following the first of the these tests, but prior to the second one, could lead to a crash of the server. (Bug #14511533, Bug #66411)

- **Replication:** It was possible in certain cases—immediately after detecting an EOF in the dump thread read event loop, and before deciding whether to change to a new binary log file—for new events to be written to the binary log before this decision was made. If log rotation occurred at this time, any events that occurred following EOF detection were dropped, resulting in loss of data. Now in such cases, steps are taken to make sure that all events are processed before allowing the log rotation to take place. (Bug #13545447, Bug #67929)

  References: See also Bug #16016886.

- **Replication:** It was possible for the MASTER_POS_WAIT() function to return prematurely following a CHANGE MASTER TO statement that updated the RELAY_LOG_POS or RELAY_LOG_NAME. This could happen because CHANGE MASTER TO did not update the master log position in such cases, causing MASTER_POS_WAIT() to read an invalid log position and to return immediately.

  To fix this problem, the master log position is flagged as invalid until the position is set to a valid value when the SQL thread reads the first event, after which it is flagged as valid. Functions such as MASTER_POS_WAIT() now defer any comparison with the master log position until a valid value can be obtained (that is, after the first event following the CHANGE MASTER TO statement has been applied). (Bug #11766010, Bug #59037)

- **Replication:** If the disk becomes full while writing to the binary log, the server hangs until space is freed up manually. It was possible after this was done for the MySQL server to fail, due to an internal status value being set when not needed. Now in such cases, rather than trying to set this status, a warning is written in the error log instead. (Bug #11753923, Bug #45449)

- **Replication:** The binary log and relay log files used the name of the PID file instead of the host name as the basename. (Bug #11753843, Bug #45359)

- **Microsoft Windows:** In Shared Memory mode, the MySQL Server could crash when receiving requests from multiple threads. (Bug #13934876)

- RPM packages were missing the innodb_engine.so and libmemcached.so plugins. (Bug #17001088)

- Windows MSI installers for MySQL 5.7 had a 5.6 upgrade code, not a 5.7 upgrade code. (Bug #16445344)

- SHOW ENGINE PERFORMANCE_SCHEMA STATUS could report incorrect memory-allocation values when the correct values exceeded 4GB. (Bug #16414644)

- Performance Schema statement tokenization overhead was reduced. (Bug #16382260)

- A long database name in a GRANT statement could cause the server to exit. (Bug #16372927)

- The server could exit if a prepared statement attempted to create a table using the name of an existing view while an SQL handler was opened. (Bug #16385711)

- For debug builds, checking of password constraints could raise an assertion for statements that updated passwords. (Bug #16289303)

- The BUILD-CMAKE file in MySQL distributions was updated with the correct URL for CMake information. (Bug #16328024)

- A Valgrind failure could occur if a CREATE USER statement was logged to the general query log and the old_passwords system variable was set to 2. (Bug #16300620)

- The optimizer's attempt to remove redundant subquery clauses raised an assertion when executing a prepared statement with a subquery in the `ON` clause of a join in a subquery. (Bug #16318585)

  References: This bug is a regression of Bug #15875919.

- Very small `join_buffer_size` values could cause an assertion to be raised. (Bug #16328373)

- Some aggregate queries attempted to allocate excessive memory. (Bug #16343992)

- Incorrect results were returned if a query contained a subquery in an `IN` clause which contained an `XOR` operation in the `WHERE` clause. (Bug #16311231)

- For debug builds, an assertion could be raised if a statement failed with autocommit enabled just before an `XA START` statement was issued. (Bug #16341673)

- Conversion of numeric values to `BIT` could yield unexpected results. (Bug #16271540)

- Certain legal `HAVING` clauses were rejected as invalid. (Bug #16221433)

- Fixed warnings when compiling with XCode 4.6. Fixed warnings when compiling when the `_XOPEN_SOURCE` or `isoctal` macro was already defined in the environment. (Bug #16265300, Bug #60911, Bug #12407384)

- Queries using range predicates that were evaluated using the LooseScan semi-join strategy could return duplicate rows. (Bug #16221623)

  References: This bug is a regression of Bug #14728469.

- For upgrade operations, RPM packages produced unnecessary errors about being unable to access `.err` files. (Bug #16235828)

- In the range optimizer, an index merge failure could cause a server exit. (Bug #16241773)

- For debug builds, the server could exit due to incorrect calculation of applicable indexes for a join that involved `const` tables. (Bug #16165832)

- The Performance Schema could return incorrect values for the `PROCESSLIST_INFO` column of the `threads` table. (Bug #16215165)

- Invocation of the range optimizer for a `NULL` select caused the server to exit. (Bug #16192219)

- `mysql_config --libs` displayed incorrect output. (Bug #16200717)

- For a `CREATE TABLE (... col_name TIMESTAMP DEFAULT CURRENT_TIMESTAMP ...) ... SELECT` statement for which the `SELECT` did not provide a value for the `TIMESTAMP` column, that column was set to '0000-00-00 00:00:00', not the current timestamp. (Bug #16163936)

- yaSSL did not perform proper padding checks, but instead examined only the last byte of plaintext and used it to determine how many bytes to remove. (Bug #16218104)

- Using `GROUP BY WITH ROLLUP` in a prepared statement could cause the server to exit. (Bug #16163596)

- References to the unused `SIGNAL_WITH_VIO_SHUTDOWN` macro in the `CMake` files were removed. (Bug #16066150)

- Setting the `slave_rows_search_algorithms` system variable to an inappropriate value could cause the server to exit. (Bug #16074161)

- Directory name manipulation could result in stack overflow on Mac OS X and Windows. (Bug #16066243)
- For debug builds, if the server was started with binary logging disabled, executing SHOW RELAYLOG EVENTS from within a stored procedure raised an assertion. (Bug #16043173)
- The query parser leaked memory for some syntax errors. (Bug #16040022)
- With the thread pool plugin enabled, large numbers of connections could lead to a Valgrind panic or failure of clients to be able to connect. (Bug #16088658, Bug #16196591)
- The server executed EXPLAIN FORMAT=JSON for some malformed queries improperly. (Bug #16078557)
- Performance Schema instrumentation was missing for slave worker threads. (Bug #16083949)
- If the error for a failed CACHE INDEX statement index within a stored program was processed by a condition handler, a malformed packet and “Command out of sync” error occurred. (Bug #16076180)
- The initial test database contained a dummy.bak file that prevented DROP DATABASE from working. This file is no longer included. Also, a db.opt file is now included that contains these lines:

  ```
  default-character-set=latin1
  default-collation=latin1_swedish_ci
  ```

  (Bug #16062056)
- Setting a system variable to DEFAULT could cause the server to exit. (Bug #16044655)
- SET PASSWORD and GRANT ... IDENTIFIED BY have no effect on the password of a user who is authenticated using an authentication plugin that accesses passwords stored externally to the mysql.user table. But attempts to change the password of such a user produced no warning, leading to the impression that the password had been changed when it was not. Now MySQL issues an ER_SET_PASSWORD_AUTH_PLUGIN warning to indicate that the attempt was ignored. (Bug #16072004)
- For debug builds, creating an InnoDB table in strict SQL mode that violated the maximum key length limit caused the server to exit. (Bug #16035659)
- Issuing a PREPARE statement using certain combinations of stored functions and user variables caused the server to exit. (Bug #16056537)
- Instances of #ifdef WITH_MYISAMMRG_STORAGE_ENGINE and #ifdef WITH_CSV_STORAGE_ENGINE in the server source code were removed because the CSV and MERGE storage engine plugins are mandatory. (Bug #15997345)
- A mysys library string-formatting routine could mishandle width specifiers. (Bug #15960005)
- The --character-set-server option could set connection character set system variables to values such as ucs2 that are not permitted. (Bug #15985752)
- During shutdown, the server could attempt to lock an uninitialized mutex. (Bug #16016493)
- The --default-authentication-plugin option permitted invalid plugin values, and did not always set the old_passwords system variable to a value appropriate for the named plugin. (Bug #16014394)
- For debug builds, executing a statement within a trigger or stored function that caused an implicit commit raised an assertion. (Bug #15985318)

- Under some circumstances, `mysql --secure-auth` permitted passwords to be sent to the server using the old (pre-4.1) hashing format. (Bug #15977433)
- With index condition pushdown enabled, queries for which the pushed-down condition contained no columns in the used index could be slow. (Bug #15896009)
- Table creation operations added entries to the Performance Schema `file_instances` table, but these were not always removed for table drop operations. (Bug #15927620)
- In special cases, the optimizer did not consider indexes that were applicable to query processing, resulting in potentially suboptimal execution and incorrect `EXPLAIN` output. (Bug #15849135, Bug #16094171)
- A query with an `EXISTS/IN/ALL/ANY` subquery with an `ORDER BY` clause ordering by an outer column of type `BLOB` that is not in the select list caused an assertion to fire. (Bug #15875919)

  References: See also Bug #14728142.
- Queries in the query cache involving a given table were incorrectly invalidated if a `TEMPORARY` table of the same name was dropped. (Bug #14839743)
- The optimizer could return nonmatching records for queries that used `ref` access on string data types. (Bug #14826522)

  References: See also Bug #14682735.
- Failure of `CREATE SERVER` due to a missing or read-only `mysql.servers` table resulted in a memory leak. (Bug #14781478)
- Enabling the query cache during high client contention could cause the server to exit. (Bug #14727815)
- Table names can be up to 64 characters, but the message string for the `ER_TABLE_NEEDS_REBUILD` and `ER_TABLE_NEEDS_UPGRADE` errors were truncating names longer than 32 characters. (Bug #14753226)
- Enabling the slow query log at runtime when access permissions on the log file prevented the server from writing to it caused the server to exit. (Bug #14711347)
- Table removal could fail and cause the server to exit for very long file names. (Bug #14581920)
- The server sometimes failed to respect `MAX_CONNECTIONS_PER_HOUR` limits on user connections. (Bug #14627287)
- The server could access the `DEBUG_SYNC` facility while closing temporary tables during connection shutdown, after the facility had been cleaned up, leading to an assertion being raised. (Bug #14626800)
- The optimizer could return incorrect results after transforming an `IN` subquery with aggregate functions to an `EXISTS` subquery. (Bug #14586710)
- If the optimizer calculated a row count of zero for the inner table of an outer join, it could not determine proper ordering for the following tables. (Bug #14628746)
- When a client program loses the connection to the MySQL server or if the server begins a shutdown after the client has executed `mysql_stmt_prepare()`, the next `mysql_stmt_prepare()` returns an error (as expected) but subsequent `mysql_stmt_execute()` calls crash the client. (Bug #14553380)
- Previously, if multiple `--login-path` options were given, `mysql_config_editor` ignored all but the last one. Now multiple `--login-path` options result in an error. (Bug #14551712)

- If MySQL server was started with options to enable the general query log or slow query log, but access permissions on the log file prevented the server from writing to it, the server started with an error message indicating that logging was off and that the server needed to be restarted after the problem was corrected. This was incorrect because it is also possible to set the logging variables again at runtime (without a restart) after correcting the problem. The error message now indicates this possibility. (Bug #14512467)
- For debug builds, creating a `TEMPORARY` table inside a trigger caused the server to exit. (Bug #14493938)
- `SHOW COLUMNS` on a view defined as a `UNION` of `Geometry` columns could cause the server to exit. (Bug #14362617)
- The `sha256_password_private_key_path` and `sha256_password_public_key_path` system variables indicate key files for the `sha256_password` authentication plugin, but the server failed to properly check whether the key files were valid. Now in the event that either key file is invalid, the server logs an error and exits. (Bug #14360513)
- `SET *var_name* = VALUES(*col_name*)` could cause the server to exit. This syntax is now prohibited because in `SET` context there is no column name and the statement returns `ER_BAD_FIELD_ERROR`. (Bug #14211565)
- The `COM_CHANGE_USER` command in the client/server protocol did not properly use the character set number in the command packet, leading to incorrect character set conversion of other values in the packet. (Bug #14163155)
- If the server was started with `--skip-grant-tables`, the `CREATE EVENT` and `ALTER EVENT` statements resulted in a memory leak. (Bug #14059662)
- Invoking the `FORMAT()` function with a locale and a very large number could cause the server to exit. (Bug #14040155)
- For debug builds, improper handling for `AUTO_INCREMENT` value overflow could cause the server to exit. (Bug #13875572)
- Certain plugin-related conditions can make a user account unusable:
  - The account requires an authentication plugin that is not loaded.
  - The account requires the `sha256_password` authentication plugin but the server was started with neither SSL nor RSA enabled as required by this plugin.

  The server now checks those conditions by default and produces warnings for unusable accounts. This checking slows down server initialization and `FLUSH PRIVILEGES`, so it is made optional by means of the new `validate_user_plugins` system variable. This variable is enabled by default, but if you do not require the additional checking, you can disable it at startup to avoid the performance decrement. (Bug #13010061, Bug #14506305)
- Passing an unknown time zone specification to `CONVERT_TZ()` resulted in a memory leak. (Bug #12347040)
- The obsolete `linuxthreads.txt` and `glibc-2.2.5.patch` files in the `Docs` directory of MySQL distributions have been removed. (Bug #11766326)
- `mysql_install_db` did not escape `'_'` in the host name for statements written to the grant tables. (Bug #11746817)
- The server could exit if built to permit a maximum number of indexes per table larger than 64.

In the course of fixing this problem, a -DMAX_INDEXES=*N* CMake option was added to permit building the server to support a larger maximum number of indexes per table. The default is 64. The maximum is 255. Values smaller than 64 are ignored and the default of 64 is used. (Bug #11761614)

- Source code in the mysys library for the my_malloc_lock and my_free_lock memory-locking APIs was never used and has been removed. (Bug #54662, Bug #11762107)
- mysqld_safe used the nonportable -e test construct. (Bug #67976, Bug #16046140)
- An out-of-memory condition could occur while handling an out-of-memory error, leading to recursion in error handling. (Bug #49514, Bug #11757464)
- The optimizer used loose index scan for some queries for which this access method is inapplicable. (Bug #42785, Bug #11751794)
- If a dump file contained a view with one character set and collation defined on a view with a different character set and collation, attempts to restore the dump file failed with an “illegal mix of collations” error. (Bug #65382, Bug #14117025)
- If the server failed to read errmsg.sys, it could exit with a segmentation fault. (Bug #53393, Bug #11760944)
- The REPLACE() function produced incorrect results when a user variable was supplied as an argument and the operation was performed on multiple rows. (Bug #49271, Bug #11757250)
- The output for SHOW CREATE VIEW could vary depending on the DEFINER account privileges. (Bug #34553, Bug #11747931)
- UNION type conversion could incorrectly turn unsigned values into signed values. (Bug #49003, Bug #11757005)
- If one thread was rebuilding a result for the query cache, other threads in the middle of using the previous result could fail to discard the old result properly. For debug builds, this raised an assertion. (Bug #66781, Bug #14631798)
- UNION ALL on BLOB columns could produce incorrect results. (Bug #50136, Bug #11758009)
- View access in low memory conditions could raise a debugging assertion. (Bug #39307, Bug #11749556)
- Attempts to create a trigger for which a trigger with the same action time and event already existed resulted in an ER_NOT_SUPPORTED_YET error rather than an ER_TRG_ALREADY_EXISTS error. (Bug #67357, Bug #14801721)
- Queries with many values in a IN() clause were slow due to inclusion of debugging code in non-debugging builds. (Bug #68046, Bug #16078212)

  References: See also Bug #58731, Bug #11765737.
- When a view definition contained a special character in the SEPARATOR clause of the GROUP_CONCAT() aggregate function, mysqldump created an invalid view definition that produced an error when the dump file was reloaded. (Bug #60920, Bug #12395512)
- Setting max_connections to a value less than the current number of open connections caused the server to exit. (Bug #44100, Bug #11752803)
- Some table I/O performed by the server when calling a storage engine were missing from the statistics collected by the Performance Schema. (Bug #68180, Bug #16222630)

- For debug builds, some queries with SELECT ... FROM DUAL nested subqueries raised an assertion. (Bug #60305, Bug #11827369)
- Nonspatial indexes only support exact-match lookups for spatial columns, but the optimizer incorrectly used range access in some cases, leading to incorrect results. (Bug #67889, Bug #15993693)
- If mysql is built with the bundled libedit library, the library is built as static code, to avoid linking to a different dynamic version at runtime. Dynamic linking could result in use of a different, incompatible version and a segmentation fault. (Bug #68231, Bug #16296509)
- SLEEP() produced no warning or error for NULL or negative arguments. Now it produces a warning, or an error in strict SQL mode. (Bug #67548, Bug #15859462)
- The --log-slow-admin-statements and --log-slow-slave-statements command options now are exposed at runtime as the log_slow_admin_statements and log_slow_slave_statements system variables. Their values can be examined using SHOW VARIABLES. The variables are dynamic, so their values can can be set at runtime. (The options were actually *replaced* by the system variables, but as system variables can be set at server startup, no option functionality is lost.) (Bug #59860, Bug #11766693)
- For arguments with fractional seconds greater than six decimals, SEC_TO_TIME() truncated, rather than rounding as it should have. (Bug #68061, Bug #16093024)
- MySQL failed to build if configured with WITH_LIBWRAP enabled. (Bug #67018, Bug #16342793)
- If the server was started without a --datadir option, SHOW VARIABLES could show an empty value for the datadir system variable. (Bug #60995, Bug #12546953)
- If a table had rows in the INFORMATION_SCHEMA.INNODB_CMP_PER_INDEX table, dropping the table did not remove those rows. (Bug #67283, Bug #14779330)
- Configuring with -DWITH_SSL=/path/to/openssl resulted in link errors due to selection of the incorrect libcrypto. (Bug #68277, Bug #16284051)
- ALTER TABLE *tbl_name* ADD COLUMN col_name TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP inserted 0000-00-00 00:00:00 rather than the current timestamp if the alteration was done in place rather than by making a table copy. (Bug #68040, Bug #16076089)
- If the server was started with --skip-grant-tables, ALTER USER ... PASSWORD EXPIRE caused the server to exit. (Bug #68300, Bug #16295905)
- CMake did not check whether the system zlib had certain functions required for MySQL, resulting in build errors. Now it checks and falls back to the bundled zlib if the functions are missing. (Bug #65856, Bug #14300733)
- mysql_install_db did not work in Solaris 10 sparse root zones. (Bug #68117, Bug #16197860)
- For EXPLAIN DELETE and EXPLAIN UPDATE the possible_keys column listed all indexes, not just the applicable indexes. (Bug #67830, Bug #15972078)
- The Perl version of mysql_install_db mishandled some error messages. (Bug #68118, Bug #16197542)
- Handling of SQL_CALC_FOUND_ROWS in combination with ORDER BY and LIMIT could lead to incorrect results for FOUND_ROWS(). (Bug #68458, Bug #16383173)
- The server did not enforce the port or report_port upper limit of 65,535 and truncated larger values. (Bug #67956, Bug #16035522)

- Adding an ORDER BY clause following an IN subquery could cause duplicate rows to be returned. (Bug #68330, Bug #16308085)
- If INET6_NTOA() or INET6_ATON() returned NULL for a row in a result set, following rows also returned NULL. (Bug #68454, Bug #16373973)
- If XA support was activated by multiple storage engines, the server would exit. (Bug #47134, Bug #11755370)
- A statement with an aggregated, nongrouped outer query and an aggregated, nongrouped subquery in the SELECT list could return incorrect results. (Bug #68372, Bug #16325175)
- The SQL_NO_CACHE keyword is supposed to prevent the server from checking the query cache to see whether the query result is already cached, and to prevent it from caching the query result. However, the query cache check was suppressed only if SQL_NO_CACHE was preceded and followed by space characters. (For example, the server checked the cache if the keyword was followed by a newline.) Now the parser requires that the preceding and following characters be whitespace characters, not spaces. (Bug #64164, Bug #13641256)
- With explicit_defaults_for_timestamp enabled, inserting NULL into a TIMESTAMP NOT NULL column now produces an error (as it already did for other NOT NULL data types), instead of inserting the current timestamp. (Bug #68472, Bug #16394472)
- Use of KILL to kill a statement in another session could in some cases cause that session to return an incorrect error code. (Bug #45679, Bug #11754124)
- If a column is declared as NOT NULL, it is not permitted to insert NULL into the column or update it to NULL. However, this constraint was enforced even if there was a BEFORE INSERT (or BEFORE UPDATE trigger) that set the column to a non-NULL value. Now the constraint is checked at the end of the statement, per the SQL standard. (Bug #6295, Bug #11744964)
- On Windows, the log_error system variable did not accurately reflect the error log file name in some cases. For example, if the server was started without --console or --log-error, the default is to log to *host_name*.err in the data directory, but log_error remained blank.

  Now log_error should be nonblank and reflect the log file name in all cases, on all platforms. The value is stderr if the server does not write error messages to a file and sends them to the console (standard error output) instead. In particular, on Windows, --console overrides use of an error log and sends error messages to the console, so log_error will be set to stderr. (Bug #8307, Bug #11745049)

Changes in MySQL 5.7.0 (Not released, Milestone 10)

Note

This is a milestone release, for use at your own risk. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running mysql_upgrade. For example, you may find it necessary to dump your data with mysqldump before the upgrade and reload it afterward.

Functionality Added or Changed

- **Important Change; Replication:** SHOW SLAVE STATUS when run concurrently with STOP SLAVE can take a long time to execute if the slave SQL thread was in the midst of applying a large update. To fix this problem, a new NONBLOCKING option has been added to the SHOW SLAVE STATUS statement.

When this option is used, SHOW SLAVE STATUS does not wait on the SQL or I/O threads but returns immediately. This means that the reported states of these threads may not be completely up to date when the option is used. NONBLOCKING is intended primarily for use by monitoring tools in which obtaining an immediate response is more important than having the most timely data. (Bug #15993588, Bug #67879)

- **Important Change; Replication:** Added the --idempotent option for mysqlbinlog, which causes the MySQL Server to employ idempotent mode. This causes suppression of all duplicate-key and key-not-found errors when processing updates from the binary log. The mode is in effect for the current mysqlbinlog client and client session only.

- **Important Change:** INSERT DELAYED is no longer supported. The server recognizes but ignores the DELAYED keyword, handles the insert as a nondelayed insert, and generates an ER_WARN_LEGACY_SYNTAX_CONVERTED warning. ("INSERT DELAYED is no longer supported. The statement was converted to INSERT."). Similarly, REPLACE DELAYED is handled as a nondelayed replace. The DELAYED keyword will be removed in a future release.

  In addition, several DELAYED-related options or features were removed:

  - The --delayed-insert option for mysqldump.

  - The COUNT_WRITE_DELAYED, SUM_TIMER_WRITE_DELAYED, MIN_TIMER_WRITE_DELAYED, AVG_TIMER_WRITE_DELAYED, and MAX_TIMER_WRITE_DELAYED columns of the Performance Schema table_lock_waits_summary_by_table table.

    If you upgrade to this release of MySQL from an earlier version, you must run mysql_upgrade (and restart the server) to incorporate these changes into the performance_schema database.

  - mysqlbinlog no longer writes comments mentioning INSERT DELAYED.

- **Microsoft Windows:** Windows Vista, Windows Server 2008, and newer support native symlinking using the mklink command. This makes the MySQL Server implementation of database symbolic links using .sym files redundant, so that mechanism is now removed. This change has the following implications:

  - Existing .sym files are now ignored. Database symlinks should be recreated using mklink. See Using Symbolic Links for Databases on Windows.

  - The --symbolic-links and --skip-symbolic-links options and the have_symlink system variable now are meaningful only for Unix systems, and not for Windows.

- The server now issues a warning if an index is created that duplicates an existing index, or an error in strict SQL mode. (Bug #37520, Bug #11748842)

- Previously, **Control+C** in mysql interrupted the current statement if there was one, or exited mysql if not. Now **Control+C** interrupts the current statement if there was one, or cancels any partial input line otherwise, but does not exit. (Bug #66583, Bug #14554568)

- The mysql_clear_password cleartext client-side authentication plugin is intended for authentication schemes that require the server to receive the password as entered on the client side, without hashing. Because the password is sent in the clear, this plugin should be used within the context of a secure connection, such as an SSL connection, to avoid exposing the password over the network. To make inadvertent use of this plugin less likely, it is now required that clients explicitly enable it. This can be done several ways:

  - Set the LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN environment variable to a value that begins with 1, Y, or y. This enables the plugin for all client connections.

  - The mysql, mysqladmin, and mysqlslap client programs support an --enable-cleartext-plugin option that enables the plugin on a per-invocation basis.
  - The mysql_options() C API function supports a MYSQL_ENABLE_CLEARTEXT_PLUGIN option that enables the plugin on a per-connection basis. Also, any program that uses libmysqlclient and reads option files can enable the plugin by including an enable-cleartext-plugin option in an option group read by the client library.

- MySQL now supports stacked diagnostics areas. When a push to the diagnostics area stack occurs, the first (current) diagnostics area becomes the second (stacked) diagnostics area and a new current diagnostics area is created as a copy of it. Within a condition handler, executed statements modify the new current diagnostics area, but GET STACKED DIAGNOSTICS can be used to inspect the stacked diagnostics area to obtain information about the condition that caused the handler to activate, independent of current conditions within the handler itself. (Previously, there was a single diagnostics area. To inspect handler-activating conditions within a handler, it was necessary to check this diagnostics area before executing any statements that could change it.) See GET DIAGNOSTICS Syntax, and The MySQL Diagnostics Area.

Bugs Fixed

- **Important Change:** Formerly, the ExtractValue() and UpdateXML() functions supported a maximum length of 127 characters for XPath expressions supplied to them as arguments. This limitation has now been removed. (Bug #13007062, Bug #62429)
- **InnoDB:** During an online DDL operation, a duplicate key error could be incorrectly issued if a record was inserted and subsequently updated while the table was being rebuilt. (Bug #14723456)
- **InnoDB:** Creating and altering tables repeatedly would result in a memory leak that was due to a duplicate key error. The duplicate key error occurred because the row_merge_build_indexes function did not call row_fts_psort_info_destroy often enough. As full-text search indexes were created with a unique index, the unique index would fail due to the duplicate key error, and full-text search build resource would not be released. (Bug #14759111)
- **InnoDB:** If the server crashed at a precise moment during an ALTER TABLE operation that rebuilt the clustered index for an InnoDB table, the original table could be inaccessible afterward. An example of such an operation is ALTER TABLE ... ADD PRIMARY KEY The fix preserves the original table if the server halts during this operation. You might still need to rename the .ibd file manually to restore the original table contents: in MySQL 5.6 and higher, rename from #sql-ib$*new_table_id*.ibd to *table_name*.ibd within the database directory; prior to MySQL 5.6, the temporary file to rename is *table_name*#1 or #2. (Bug #14669848)
- **InnoDB:** InnoDB IO threads within Performance Schema were exposed with the following name: “io_handler_thread”. This fix implements specific keys such as io_read_handler_thread, io_write_handler_thread, io_ibuf_handler_thread to differentiate InnoDB IO threads within Performance Schema. (Bug #14670810)
- **InnoDB:** Inserting data of varying record lengths into an InnoDB table that used compression could cause the server to halt with an error. (Bug #14554000, Bug #13523839, Bug #63815, Bug #12845774, Bug #61456, Bug #12595091, Bug #61208)
- **InnoDB:** This fix addresses an assert condition that would occur when inserting large BLOBs into tablespaces with a 4KB physical page size or into some compressed tables. Extents would not be allocated soon enough for tablespaces with smaller physical page sizes. (Bug #14520559)
- **InnoDB:** If a table was defined with an index key length very close to the upper length limit of 3072, a query against that table could cause a serious error. (Bug #14500557, Bug #14537695)

- **InnoDB:** In debug builds, a mismatch in the InnoDB PAGE_FREE list would cause an assertion. (Bug #12701488)
- **InnoDB:** On Linux systems, certain I/O requests that read or wrote fewer than the requested number of bytes could cause the server to crash. This issue could happen more frequently with asynchronous I/O requests. The messages did not clearly identify what type of error occurred:

  ```
  InnoDB: Operating system error number 0 in a file operation.
  InnoDB: Error number 0 means 'Success'.
  ```

  With this fix, MySQL retries the operation several times before giving up. (The number of retries is defined by the constant NUM_RETRIES_ON_PARTIAL_IO in the source code, default value 10.) (Bug #11761646, Bug #54160)
- **Partitioning:** When the server is started with --skip-partition, it should reject DDL or DML statements on partitioned tables. However, for DROP TABLE, the server dropped the .frm file, and for RENAME TABLE, the server renamed the .frm file. (Bug #11763795)
- **Replication:** mysqlbinlog did not properly decode DECIMAL values in a row-based binary log. This could cause invalid values to be printed out for DECIMAL columns. (Bug #14309019)

  References: See also Bug #17544169.
- **Replication:** When using mysqlbinlog with the --verbose option to read a binary log written by a MySQL server using row-based or mixed-format logging, invalid SQL could be produced when comments appeared inside BINLOG statements. One way in which this could happen was when a function that updated data was used within an INSERT ... SELECT statement. (Bug #12889121)
- **Replication:** mysql_upgrade on the master broke replication when the slave was run with --log-output equal to FILE or NONE. (Bug #11763447)
- **Replication:** Issuing STOP SLAVE caused a spurious Error reading packet from server: Lost connection to MySQL server during query message to be written to the error log. (Bug #11761457, Bug #12977988, Bug #53955)
- **Replication:** When an error occurs in the slave SQL thread, this causes the Slave_SQL_Error and Slave_SQL_Errno columns from SHOW SLAVE STATUS to display the reason for the error. The error number should be one of the usual constants ER_* defined in sql/share/errmsg.txt, and the error message should be the corresponding string. However, in some cases, Slave_SQL_Errno was set to something other than an ER_* number, and Slave_SQL_Error to a hard-coded error message rather than a translatable string from sql/share/errmsg.txt. Now all errors shown by SHOW SLAVE STATUS originate in sql/share/errmsg.txt, as expected. (Bug #11760365, Bug #52768)
- **Microsoft Windows:** On Microsoft Windows, CMake entries for POSIX API's not found on Microsoft Windows were added to the CMake cache. This decreases the number of expected "Not found" errors while compiling MySQL. (Bug #14790333)
- **Microsoft Windows:** On Microsoft Windows, queries referring to a table with invalid characters would search the system for invalid file names. The generated system error code (ERROR_INVALID_NAME) was not recognized by MySQL, so this unknown error would be reported to the server log as "ERRNO: 22 - INVALID ARGUMENT". MySQL now recognizes these errors and reports them as the table does not existing, and it no longer logs them to the server error log. (Bug #14642248)
- **Microsoft Windows:** On Windows, starting the server with --log-error and --console caused the server to write to the log file but not the console. Before MySQL 5.5.3, this occurred only if --log-error was specified after --console. Now, --console overrides --log-error no matter the option order so that --console produces console output in all cases. (Bug #14207773, Bug #65592)

- **Microsoft Windows:** It was possible to specify a Named Pipe that was already in use. This is no longer allowed, as an error is now emitted and the process is aborted. After mysqld.exe was started in Named Pipe mode with a pipe name that was already used by a different instance, neither instance was able to shut down properly when a shutdown command was received from a TCP socket in any of the processes. Therefore, mysqld.exe was not terminated. (Bug #13891058, Bug #61885)
- **Microsoft Windows:** On Microsoft Windows, a failed API or function call in mysqld.exe could sometimes report the error code 22, instead of the proper error code. (Bug #11763004)
- **Cluster Replication:** Transactions originating on a replication master are applied on slaves as if using AO_AbortError, but transactions replayed from a binary log were not. Now transactions being replayed from a log are handled in the same way as those coming from a "live" replication master.

  See The NdbOperation::AbortOption Type, for more information. (Bug #14615095)
- Joins of exactly 32 tables and containing a HAVING clause returned an empty result. (Bug #15972635)
- The parser rejected some legal UNION statements. (Bug #14730856)
- Setting thread_cache_size to a negative value at server startup resulted in a value of 16384 rather than 0. (Bug #14683107)
- The return value from IS_USED_LOCK() was reported using the wrong data type. (Bug #14575699)
- There was no warning at startup if the server was started with an invalid query_cache_size value. (Bug #14576423)
- Activation of a stored program handler did not preserve the current diagnostics stack. (Bug #14342913)
- In debug builds, killing a HELP statement caused an assertion to be raised. (Bug #14221840)
- If an error occurred during evaluation of the BEFORE expression of a PURGE BINARY LOGS BEFORE statement, the statement did not abort as it should have and later raised an assertion. (Bug #14215847)
- For the index merge access method, the optimizer could make a suboptimal choice of indexes to use. (Bug #14095506)
- An assertion could be raised if the attempt to open the .frm file for a temporary table failed. (Bug #13359247)
- If the state of the Event Scheduler was changed during server shutdown, the server could crash. (Bug #13002460)
- When storing the definition for a view that used the UPPER() or LOWER() function, the function call was replaced by UCASE() or LCASE(), respectively (as shown in the output of SHOW CREATE VIEW). This was in spite of the fact that UPPER() and LOWER() are standard, with UCASE() and LCASE() being MySQL synonyms for these. This made it more difficult to move databases between MySQL and other database systems.

  With this fix, calls to UPPER() and LOWER() within views are no longer rewritten when storing their definitions; instead, UCASE() is now rewritten as UPPER() in stored view definitions, and LCASE() as LOWER(), which increases the portability of the views. (Bug #12844279)
- For queries that accessed an INFORMATION_SCHEMA table in a subquery, an attempt to lock a mutex that had already been locked could cause a server crash. (Bug #11765744)
- The Range checked for each record optimization is now used for conditions with outer query references. (Bug #11750963)

- An INSERT INTO ... SELECT statement that inserted no rows unnecessarily invalidated statements in the query cache that used the target table. (Bug #50065, Bug #11757947)

- The server refused client connections while executing FLUSH PRIVILEGES. (Bug #63178, Bug #13418619)

- References to a stored function without a database name qualifier while there was no default database resulted in an ER_SP_DOES_NOT_EXIST error rather than ER_NO_DB_ERROR. (Bug #64692, Bug #13864485)

- Concurrent inserts were blocked by selects if the inserts were generated from within a stored procedure. (Bug #58689, Bug #11765698)

- Concurrent execution of DROP DATABASE and any of CREATE FUNCTION, CREATE PROCEDURE, or CREATE EVENT could be written to the binary log in the wrong order, causing replication failure. (Bug #65428, Bug #14127220)

- INSERT INTO ... SELECT ... ON DUPLICATE KEY UPDATE and LOAD DATA CONCURRENT REPLACE took too weak a lock, leading to the possibility of concurrent SELECT statements returning inconsistent results. (Bug #38046, Bug #11749055)

- If read_only is enabled, it is still permitted to create TEMPORARY tables. But in this case, a non-TEMPORARY table with the same name could also be created, which should not be permitted. (Bug #64992, Bug #13969578)

- Enabling the session value of low_priority_updates had no effect for INSERT statements. (Bug #64892, Bug #13939940)

- An event was not dropped from the mysql.event table under these circumstances: The event was created while the event scheduler was enabled; the scheduler was disabled and re-enabled; the event expiration time was reached. (Bug #34804, Bug #11748012)

- Using ALTER TABLE to rename a table to . resulted in a table with no name. (Bug #49636, Bug #11757569)

- SHOW CREATE VIEW failed if the tables underlying the view were changed. (Bug #61718, Bug #12762393)

- SHOW GLOBAL STATUS caused performance problems on busy servers due to lock contention. (Bug #42930, Bug #11751904)

- For a view defined on a UNION, the server could create an invalid view definition. (Bug #65388, Bug #14117018)

- For queries using ref access on string data types, the ref access condition could be evaluated again as part of the query condition or pushed down as an index condition to the storage engine. (Bug #66983, Bug #14682735)

- For an ALTER TABLE statement that renamed or changed the default value of a BINARY column, the alteration was done using a table copy and not in place. (Bug #67141, Bug #14735373, Bug #69580, Bug #17024290)

- A view was created with an incorrect definition if the WHERE clause contained string literals and character_set_client and character_set_connection were set to different character sets. (Bug #63094, Bug #13520710)